Sommario

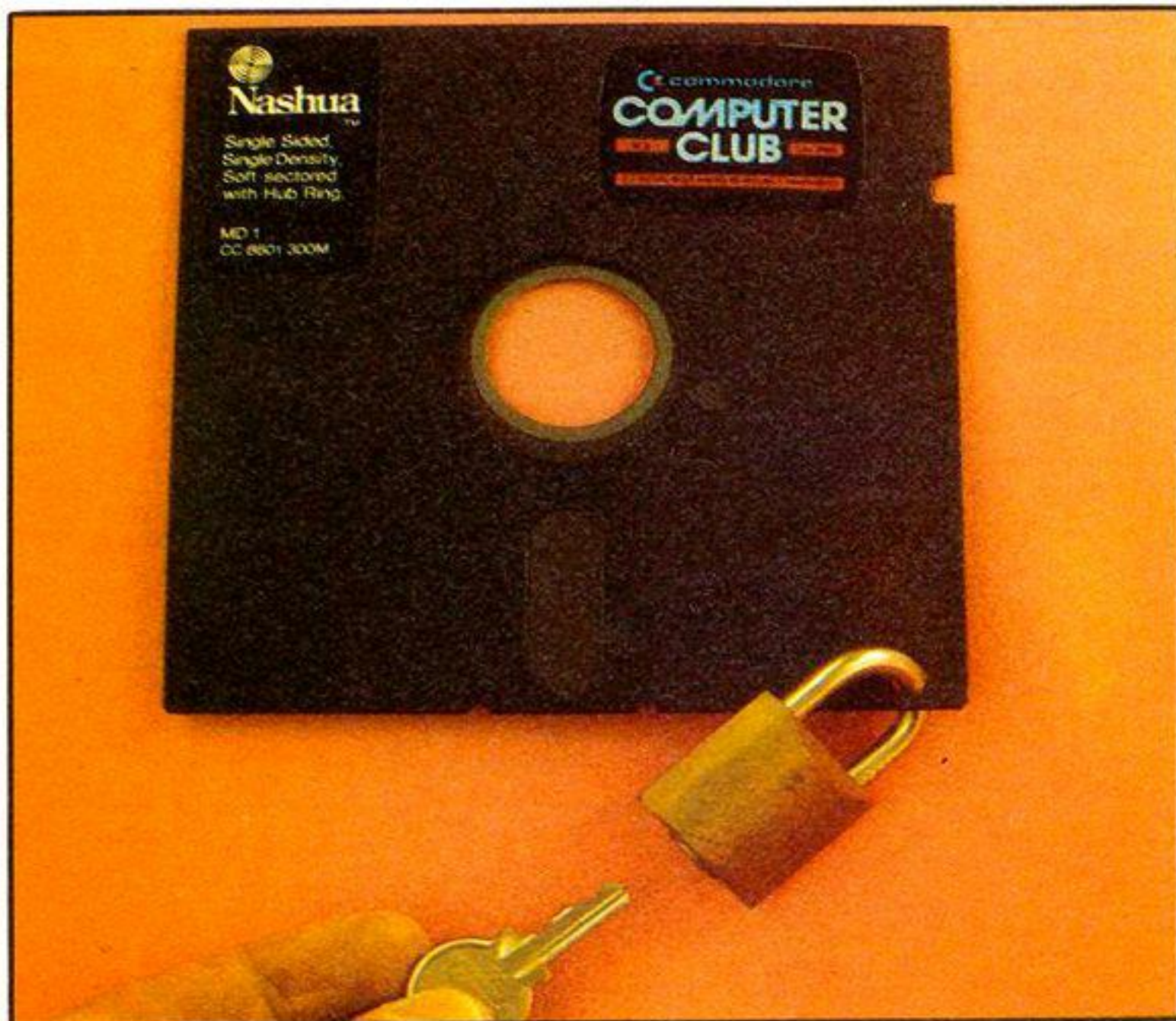
RUBRICHE

L'immagine di copertina è stata tratta dalla locandina del film: "Mister Crocodile Dundee II"

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Baiocchi, Angelo Bianchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Mietta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma
Grafica: Arturo Ciaglia, Elena Salvadori
Direzione, redazione, pubblicità: v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Giorgio Ruffoni - Viale Famagosta, 75 - 20142 Milano
Tel. 02/8467348
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Tiziana Sodano - **Abbonamenti:** Liliana Spina
Tariffe: prezzo per copia L. 4.500. Abbonamento annuo (11 fascicoli) L. 45.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 85.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70%
Distrib.: MePe - via G. Carcano, 32 - Milano
Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age -Energy Manager - Hospital Management - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - VR Videoregistrare

CODICE PRIVATO

Il recente film con Ornella Muti assegna, ad un computer, il ruolo di interprete di secondo piano; ma solo in apparenza.

di **Alessandro de Simone**

Non è certo abitudine della nostra rivista parlare di film nè, tantomeno, di "fare" critica cinematografica.

L'ultimo film di Francesco Maselli, tuttavia, offre lo spunto per dimostrare che un computer può far parte della vita di tutti noi, nella vita di tutti i giorni.

La storia vede impegnata una bravissima Ornella Muti nel difficile ruolo di unica protagonista (non appare, infatti, nessun altro nel corso dell'intero svolgimento della vicenda) che, abbandonata dal suo uomo senza un apparente perchè, tenta in tutti i modi di scoprire il motivo di questo improvviso ed inaspettato abbandono.

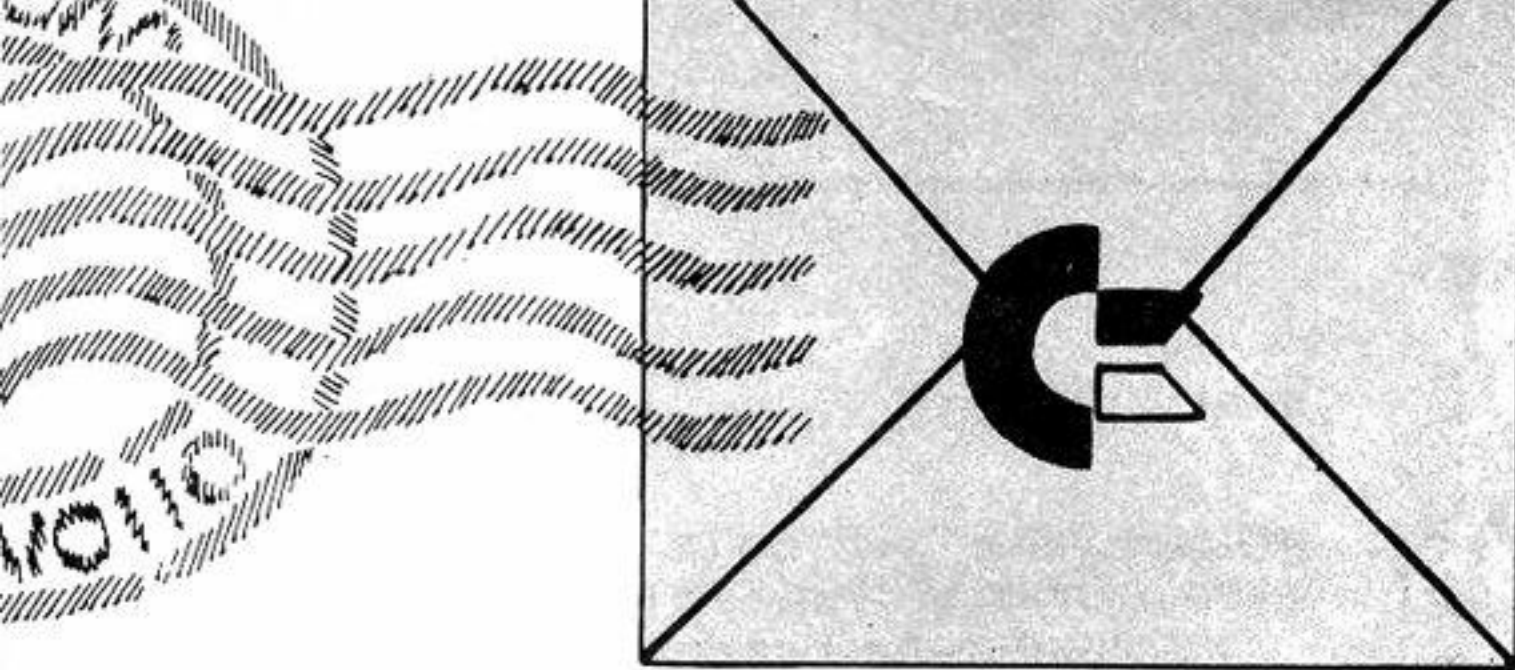
L'intero film si svolge all'interno di un enorme superattico ove, tra i vari elementi di arredo, sono sistemati, in modo da essere sempre pronti per l'uso, alcuni strumenti tecnologici, attualmente molto comuni. Ci riferiamo al telefono (un inquietante apparecchio nero, di vecchia fattura, che nel finale lascia di stucco anche lo spettatore abituato a colpi di scena), un sistema di proiezione incrociata di diapositive, un videoregistratore e, naturalmente, un computer.

Ed è proprio quest'ultimo apparecchio, ed il suo ruolo svolto nel film, che colloca la pellicola al di fuori degli schemi tradizionali. Non è, infatti, uno di quegli osceni calcolatori tutto-fare dei cartoni animati giapponesi, nè quell'aggregato di led lampeggianti che la filmografia deteriore ci ha abituato a ingurgitare. Non è nemmeno un elaboratore dalle potenzialità semi-umane, in grado di prender decisioni autonome ed "intelligenti" (2001 Odissea nello spazio), addirittura antropomorfo (Alien) o, ancora, fantapolitico (Wargames); pellicole che, tuttavia, hanno regalato momenti di intensa emozione agli appassionati di questo genere.

Si tratta di un "banale" personal computer, e nemmeno di quelli più rari, dotato, per di più, di programmi comuni ed utilizzabili da chiunque. Quando Anna (Ornella Muti) intuisce, infatti, che una parte della verità cercata può esser nascosta nel calcolatore (usato abitualmente dal suo uomo) non fa altro che ripetere meccanicamente, ad alta voce, le istruzioni apprese (..."spostare la freccia sull'opzione e premere il tasto"...) per entrare nel word processor, nel data base e negli altri archivi più o meno vasti contenuti nella memoria del computer.

Ad un certo punto si trova nell'impossibilità di proseguire la ricerca perchè viene richiesta la digitazione di un inaspettato "codice privato". Il bello del film (e per questo inviamo un grazie sentitissimo al suo regista) consiste nel fatto che Anna non ricorre a squallide, quanto improbabili, tecniche sofisticate per la determinazione del codice, di cui sono infarciti i telefilm di mezza tacca. Il codice, molto più modestamente (ed "umanamente"), viene facilmente scoperto grazie all'intuito della protagonista che si ricorda della presenza di una busta, priva di francobollo, custodita in cassaforte; viene anzi il sospetto che il suo uomo l'abbia intenzionalmente aiutata, offrendole un'ovvia soluzione al problema.

Ed è propro questo uno dei motivi per cui abbiamo apprezzato il film; il computer viene infatti presentato non più come un qualcosa di misterioso, se non di ostile. Assume, nel film di Maselli, un ruolo più familiare, che esce dall'eccezionale (strumento di un'astronave, terminale segreto di una base militare, eccetera), per entrare nel "quotidiano", alla portata di tutti, perfino della protagonista che riconosce, senza vergogna, di aver completato a malapena la scuola dell'obbligo. Al personal computer, insomma, viene finalmente restituito il ruolo di strumento domestico, utile per archiviare, indifferentemente, sia le spese di casa che le lettere segrete; i numeri di telefono come varie immagini digitalizzate. Oppure i propri intimi pensieri, come un diario elettronico dalle inconsuete potenzialità. Viene quindi delineata una interessante interpretazione dello "strumento" computer che risulta, in fin dei conti, paragonato al più noto telefono, al proiettore per diapositive, al videoregistratore ai quali affidare non solo parole ed immagini, ma anche la funzione di "chiave" di lettura delle immagini stesse e dei pensieri, diversa da persona a persona; un codice personale.

"Privato", appunto.

la vostra posta

TERRORE A DUNWICH

□ Potete fornire la soluzione del vostro favoloso gioco "Terrore a Dunwich"?
(Emanuele M. - Salvatore T. - Marco Rinaldi - Tiziano Gogna - Fabio Risi ed altri lettori)

• Michele Maggi, il responsabile della pubblicazione "parallela" sulla quale vengono proposti, periodicamente, numerosi giochi di un certo pregio, si è rifiutato di comunicarmi la soluzione, per ovvi motivi; ebbene sì, lo ammetto: nemmeno io sono riuscito a completare il gioco. Maggi mi ha comunque assicurato che, trascorso un po' di tempo, divulgherà tutte le soluzioni delle varie avventure, una per una. Si tratta, quindi, di avere pazienza e, nel frattempo, continuare nei tentativi...

LAMENTELE

□ L'adattatore telematico 6499 per C/64 sembra non esser compatibile con il C/128 (usato, ovviamente, in modo 64). E' possibile una cosa del genere?
(Luca Mastini - Opera)

• Sembra che, purtroppo, il nostro lettore abbia ragione. Non rimane che prendere atto della lagnanza ed annotarla nel libro delle incompatibilità tra C/128 (usato in modo 64) e C/64 "puro".

Per collegare via telefono il C/128 alle varie banche dati, quindi, consigliamo di applicare uno dei tanti modem in commercio rivolgendosi al florido mercato dell'area milanese, cui appartiene la tua cittadina.

EPPUR MI PARLA

□ In una precedente risposta avete asserito che il C/16 può "parlare" a patto di possedere un'espansione di memoria. Io posseggo il C/16 inespanso su cui gira il gioco "Terra Nova" che contiene una frase digitalizzata in modo impeccabile. Chi ha ragione?
(Claudio Lanzoni - M. Lombarda)

• Tutti e due; cioè sia tu (che possiedi una prova tangibile di quanto asserito) sia A. Diano che ha fornito la risposta accennata. In questa, infatti, si voleva solo minimizzare la potenzialità del C/16 inespanso, che può gestire la pagina grafica OPPURE fare altre cose impegnative.

In linea teorica un qualsiasi computer che disponga di emissione sonora appena decente può "parlare". Si tratta di studiare esattamente le sue potenzialità sonore, da sfruttare per mezzo di una miriade di byte, scaturiti da un digitalizzatore sonoro, che coraggiose software house utilizzano per lo scopo prefissato.

Si tratta, quindi, di riversare su un computer le varie migliaia di byte elaborati, a parte, altrove. Non si può pretendere, insomma, di realizzare su C/16 (addirittura inespanso!) un Tool vocale che consenta comandi tipo :Say "frase digitalizzata" o simili come, ad esempio, è possibile fare con il C/64. Con quest'ultimo, in ogni caso, i vari codici che riproducono il piccolo miracolo derivano anch'essi da studi portati a termine grazie ad efficaci digitalizzatori.

E SE POI NON MI PIACE?

□ Terminati gli studi (liceo scientifico) vorrei studiare Informatica all'università. Ho paura, però, di scoprire di non esser portato per queste discipline.
(Dino Calvisi - Petogna)

• "Del futur non v'è certezza" diceva qualcuno tempo addietro, che però, per consolare, suggeriva "Chi vuol esser lieto, sia".

Se, però, sei appassionato ORA di informatica, non riesco a capire per quale motivo tu debba perdere interesse o, peggio, non riuscire nel campo professionale.

Del resto, per quello che mi pare di capire dalla lettera, c'è ancora tempo prima di prendere la decisione. Per il momento coltiva il tuo hobby con impegno e sforzati di applicarti il più possibile, soprattutto nel campo del linguaggio macchina.

INTERRUZIONI INSPIEGABILI

□ Tentando, con il mio C/128-D, di memorizzare sotto forma di file sequenziale il contenuto di un migliaio di locazioni di memoria, il drive all'improvviso si blocca senza un motivo apparente.
(Claudio Castellini - Genova)

• Le indicazioni fornite, purtroppo, sono troppo scarse; sarebbe stato più utile inviarci il programma al cui interno è presente la procedura descritta.

Dalla lettera, comunque, ho il sospetto che le altre routine l.m. (anch'esse appartenenti allo stesso programma) possano influenzare negativamente il funzionamento delle operazioni I/O.

Una cosa è certa: la routinetta accennata è priva di errori ed il malfunzionamento va certamente ricercato tra insolite operazioni che altrano, in tutto o in parte, locazioni vitali del Basic.

BUG IN GW-BASIC

□ Ritengo di aver individuato alcuni "bug" nel vostro prodotto Gw-Basic Emulator (V.2) che vorrei segnalare (segue elenco).
(Andrea ponzio - Collegno)
(Giancarlo Perdua - Voghera)

• Chiariamo i dubbi del lettore, uno per uno:

1- Il comando Verify non esiste nel Gw-Basic "originale" e, di conseguenza, non è stato implementato nell'emulatore. Il comando, tuttavia continua a funzionare perfettamente nel C/64 dotato di Gw-Basic, a patto di assegnare, con precisone, il nome riscontrabile in "ambiente" Commodore.

Ciò significa, in altre parole, che è necessario far precedere il nome dai cinque spazi bianchi o dai codici delle subdirectory cui appartiene il file da verificare. Per evitare errori, quindi, consigliamo di visualizzare l'intera directory del dischetto con il comando "Dirall" e di digitare, per intero ed esattamente come appare sul video, il nome del programma da verificare.

2- Con il Gw-Basic è ancora possibile usare i tasti Shift e Commodore per passare dal set maiuscolo - grafico al maiuscolo - minuscolo. Tuttavia, per evitare errori di digitazione, l'emulatore tende sempre a riportare il set maiuscolo-grafico. E' ciò che succede, appunto, impartendo il comando List che, alla fine, ripristina il set maiuscolo - grafico.

3- L'articolo "Relativamente utili" pubblicato sul N. 40 di C.C.C. (marzo '87) si riferiva alla sintassi della prima versione Gw-Basic Emulator. La nuova versione, in circolazione da ottobre '87, è dotata di una sintassi più vicina al Gw-Basic ufficiale e, di conseguenza, incompatibile con il vecchio programma citato.

Ne approfittiamo per ricordare che altri comandi sono stati modificati rispetto alla versione precedente, come indicato chiaramente sul libretto di istruzioni allegato al dischetto "Ms-Dos Emulator".

NOVITA' E CURIOSITA'

☐ Quale è la novità software per C/128 cui accennavate nel N. 47? Come realizzate la stupenda musica che è possibile ascoltare nei vostri prodotti su disco e nastro?
(Francesco Gringoli - Padova)

• Ci riferivamo alla possibilità di leggere (con C/128-D oppure C/128 + drive 1571; con il 1541 non funziona) dischetti contenenti programmi in Gw-Basic provenienti da un "vero" computer Ms-Dos e adattarli in un formato leggibile, in modo 64, con il nostro Gw-Basic emulator. Per saperne di più chiedi il dischetto al nostro servizio arretrati.

La musica che ascolti in sottofondo è prodotta da un bravissimo estimatore di musica che ci rifornisce, periodicamente, di quasi tutti i brani che tu, giustamente, apprezzi.

Come "strumento" software viene utilizzato il nostro Music Editor per C/64 divulgato in edicola tempo addietro. Non dimenticare, però, che per raggiungere quei livelli di bravura il computer non basta: è indispensabile conoscere molto bene la musica...

SOFTWARE

MEMORIZZARE UNA SCHERMATA

☐ Come posso memorizzare una schermata in bassa risoluzione e poi richiamarla in altri listati?
(Angelo Candeloro - Avezzano)

• Abbiamo pubblicato più volte routine del genere; tuttavia, per la gioia dei principianti...

```
100 print"scrivi quello che vuoi"
110 print"poi batti: run 200"
120 stop
200 open 1,8,8,"schermo,s,w"
210 for i=1024 to 2023
220 print# 1, peek(i): next
230 close 1
240 fori= 1 to 300: next
250 print chr$(147) "guarda..."
260 open 1,8,8,"schermo,s,r"
270 for i=1024 to 2023
280 input# 1, x
290 poke i,x: next
300 close 1
```

Se la schermata, al momento della registrazione, è già memorizzata in 1000 locazioni diverse da quelle originali del video (1024 - 2023), è sufficiente modificare la riga 210 inserendo, al posto di 1024, la locazione iniziale e, al posto di 2023, quella finale.

ESPANSIONI PER C/16

☐ Vorrei realizzare un'espansione di memoria per il mio C/16 ma sono privo dell'opportuna documentazione hardware. Esiste un testo sufficientemente approfondito al riguardo?
(Gianluca Bassini - S. Bassano)

• Purtroppo non sono in grado di aiutarti se non ricordando che sul n. 50 abbiamo parlato dell'espansione da 64 KRam per C/16, già bella e pronta, in vendita presso la ditta Niwa di Sesto San Giovanni (tel. 02 / 26.20.312).

Purtroppo non sono a conoscenza di un volume specifico per lo sfortunato computer nè ti consiglio di effettuare (pericolose) modifiche hardware.

STAMPANTE O DRIVE?

☐ Sono indeciso, a causa delle mie modeste risorse, se acqui-

stare il drive o la stampante, che mi sarebbero davvero utili.
(Luciano Riccardi - Roma)

• Devi comprare entrambe le periferiche, senza alcun dubbio. Se, infatti, vuoi procurarti un drive, vuol dire che il registratore non ti basta più; ciò significa che hai spesso a che fare con programmi impegnativi che non ti lasciano il tempo di trastullarti con il datassette.

I programmi impegnativi, però, (sia in Basic che in l.m.) sono lunghi, e non è possibile esaminarli con calma servendosi del solo schermo. Ergo: occorre anche la stampante.

E' solo una questione di denaro; per procurartelo fai leggere queste righe ai tuoi genitori.

Cari genitori di Luciano, ci vogliamo decidere ad accontentare questo bravo figliolo che non chiede altro che di impegnarsi in un'attività che, oggi, è un hobby ma, domani, può essere una buona professione?

(A' Lucià: poi famo a metà...)

MESSAGGI IN KOALA

□ E' possibile inserire messaggi nelle schermate Koala?
(Dino Calvisi - Petogna)
(Donatello Rizzo - Melissano)

• Il programma "Koala" è stato il primo vero programma grafico per C/64 che consentiva di attivare una quantità (per quei tempi) incredibile di funzioni: cerchi, linee, spezzate, scambi di pagina grafica, spostamenti di porzioni di disegno, cambio di colori e così via.

Il programma era già sufficientemente "miracoloso" ed i suoi autori non pensarono ad inserire una (banale) routine per aggiungere messaggi.

Purtroppo non sono disponibili programmi in grado di apportare modifiche al Koala, ma sono in circolazione numerosi package grafici, per C/64, sufficientemente sofisticati che sono in grado di svolgere la funzione richiesta.

NON FUNGE

□ Vorrei suggerire una semplice modifica da apportare al programma "Un detective grafico", pubblicato sul fascicolo di giugno, per esaminare eventuali schermate presenti sotto le Rom.
(Lord - Rieti)

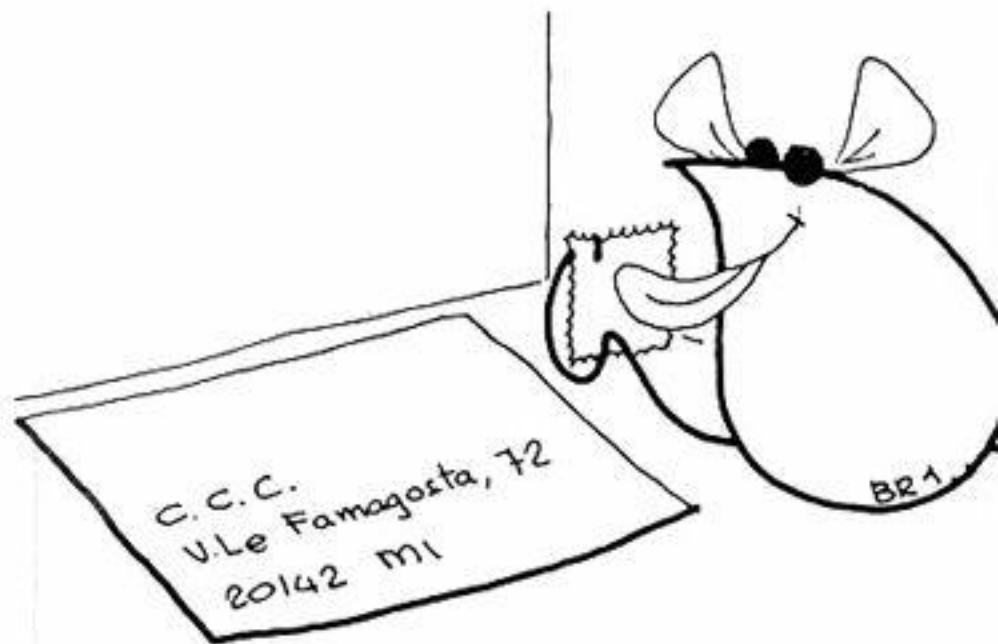
• Ho apportato la modifica, ma il programma si blocca segnalando "Out of memory error". Sei sicuro di non aver dimenticato nulla?

RESET, MORTE E DISTRUZIONE

□ Un mio amico ha affermato che se uso spesso il tasto di Reset, che io stesso ho costruito per il mio vecchio C/64, rischio di far "saltare" tutto perchè il microprocessore non ne prevede il collegamento. Inoltre vorrei delucidazioni sulla batteria tampone per l'espansione Amiga.
(Alessandro Gennari - Cavallino)

• Guardati da certi amici e telefona subito ai Carabinieri nel caso dovessi incontrarne altri che divulgano simili notizie farneticanti e sovversive.

Un QUALUNQUE microprocessore, infatti, DEVE possedere, tra i vari pin (= piedini) quello di reset.

Quando si "accende" il computer, cioè si fornisce la tensione elettrica, questa attiva i vari componenti che, per vari motivi, raggiungono lo stato di massima efficienza dopo alcuni decimi di secondo e, per di più, mai in contemporanea tra loro. Lo stesso microprocessore, quando riceve tensione, non attiva nello stesso momento tutte le decine di migliaia di transistor di cui è costituito. Nel momento in cui raggiunge le condizioni di regime, è molto probabile che ogni transistor stia andando per conto suo. E' quindi necessario che un "qualcosa" provveda non solo a mettere a posto le varie priorità all'interno del processore ma, soprattutto, gli comunichi quale è la routine (residente su ROM) che deve attivare per prima.

A questo compito provvede un particolare circuito elettrico progettato per raggiungere la massima tensione dopo un intervallo di tempo maggiore di quello che occorre ai vari chip per raggiungere la "saturazione"; ciò permette di esser sicuri, che quando detto circuito raggiunge il massimo, il sistema è pronto per l'operazione descritta che è, appunto, chiamata di "Reset".

Se, quindi, durante il normale funzionamento, si "manipola" (mediante il pulsante che hai costruito) il pin di reset non si fa altro che ripetere la stessa identica operazione eseguita dal computer al momento dell'accensione. Anzi, ricorrendo alla pressione del tasto reset, invece che allo spegnimento e successiva riaccensione, si evita di far subire stress (elettrici) al computer.

Per ciò che riguarda la batteria tampone presente nell'espansione di memoria per Amiga, tieni presen-

te che la scheda è progettata per funzionare anche per lunghi periodi di inattività. Nessun pericolo, quindi, di eventuali malfunzionamenti in caso di limitati periodi d'uso.

COME FUNZIONANO?

□ Come funzionano i comandi Load e Verify?

(Luca Cassioli - S.A. Romano)

• Dopo che un programma è stato registrato su nastro o disco rimane ancora, ovviamente, in memoria Ram. Con l'operazione Verify si controlla che, in effetti, ciò che ora si trova su supporto magnetico corrisponda, byte per byte, a ciò che si trova in memoria Ram.

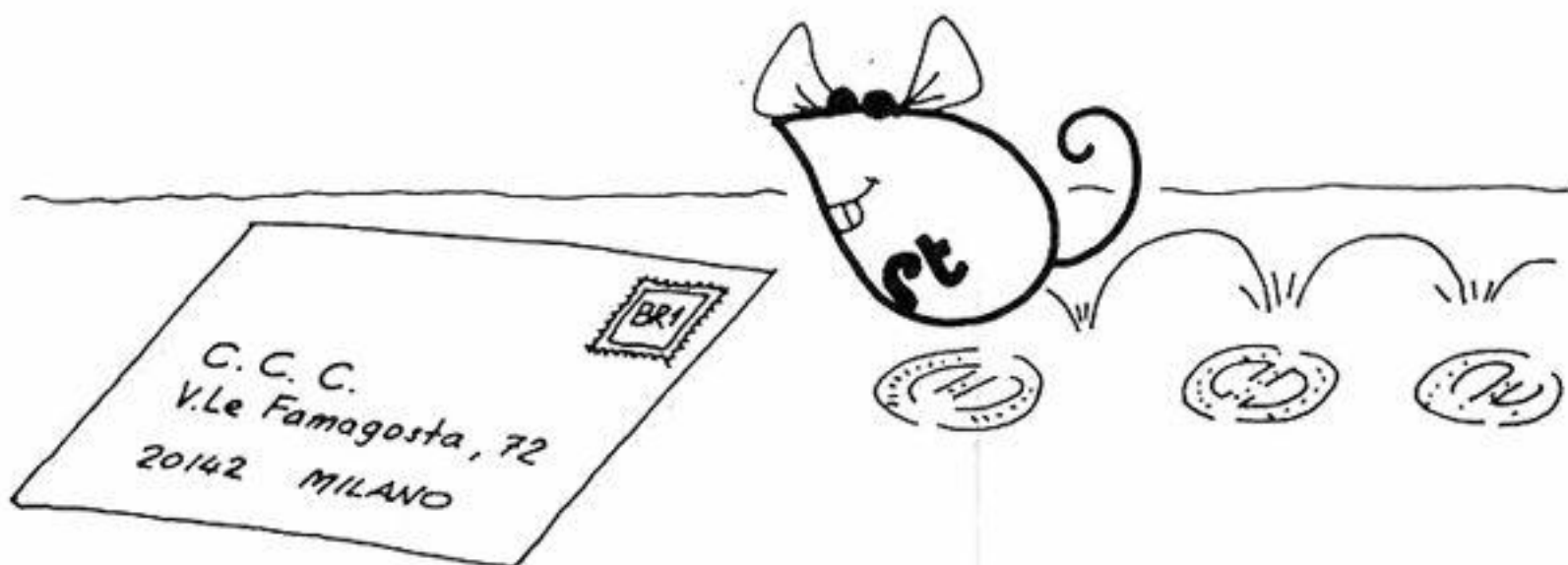
Con il comando Verify, pertanto, viene attivata una procedura che rilegge tutti i byte registrati, uno alla volta (senza, cioè, depositarli permanenetemente in altra zona Ram), e li confronta con quelli corrispondenti in memoria. Se l'operazione viene portata a termine con successo compare il messaggio "OK" altrimenti "Verify error".

Operando con il registratore la procedura è più lunga non solo per la lentezza tipica della periferica, ma anche perchè, al momento della registrazione, il programma viene registrato due volte, per motivi di sicurezza. Il nastro cassetta, infatti, era (ed è) ritenuto poco affidabile. Una "doppia" registrazione automatica, seguita per di più dall'operazione di Verify, consente una percentuale minima di errori.

Con un registratore, al momento di un Load, il programma viene quindi letto due volte: durante la prima fase i singoli byte vengono depositati in memoria; durante la seconda fase ciascun byte della seconda registrazione viene confrontato con quanto già depositato in memoria. In caso di discordanza compare "Load error".

Con opportune tecniche, quindi, è possibile, in fase di Load, interrompere la lettura a metà caricamento, a patto di conoscere con certezza il momento opportuno in cui premere il tasto Run / Stop. Subito dopo, evitando di fare QUALSIASI altra operazione, occorre digitare:

```
Poke 43, Peek (829)
Poke 44, Peek (830)
Poke 45, Peek (831)
Poke 46, Peek (832)
Clr: Restore
```

Ripetiamo che questa procedura, con il C/64, è valida solo con operazioni di caricamento dal registratore a cassetta e con la procedura standard Commodore. Quasi tutti i Turbo Tape, infatti, eliminano la "seconda" registrazione per accelerare la procedura.

ISTRUZIONI SCORRETTE

□ Ho modificato un programma in mio possesso per "dirottare" su stampante la visualizzazione che, altrimenti, verrebbe riportata solo su video; purtroppo la procedura non funziona.

(Lorenzo Marchese - Udine)

• Purtroppo mi è pervenuta la sola versione su carta (e non su disco, come supplico di solito), ma posso assicurare che la tecnica di ricorrere al comando CMD è la meno consigliabile per effettuare operazioni del genere.

In alcuni casi tale procedura funziona, ma molto più spesso, soprattutto in presenza di caratteri semigrafici e "speciali" (Clr / home, Crsr, Del, eccetera) si manifestano malfunzionamenti di vario tipo.

Il consiglio, quindi, è quello di riscrivere la (breve) subroutine sostituendo i normali...

Print "caratteri da scrivere"

...con i più appropriati Print seguiti dal carattere cancelletto (shift + 3)...

Print#, "caratteri da scrivere"

...e limitandosi ad usare il comando CMD solo per scrivere listati, come consigliato sui manuali del computer e della stampante.

STRANI ACCADIMENTI

□ Ho deciso di non comprare più la vostra rivista perchè da un po' di tempo fa schifo, sembrate impazziti. Dei vari articoli la maggior parte sono chiacchiere, per il resto non si capisce un c... Che cosa vi è successo? (ed altri insulti random)
(Lettera anonima)

• E' successo che abbiamo deciso di rivolgerci solo ad un pubblico educato, facendo in modo di escludere tutti gli altri; la tua lettera, appunto, rappresenta un'efficace conferma.

STRANE IDEE

□ Avrei in mente di scrivere un programma contenente messaggi subliminali con lo scopo di favorire terrorismo e spionaggio informatico. Solo che ho problemi a causa della lentezza del Basic...
(Lettera quasi firmata)

• Diamine! Ci sono metodi molto più semplici e rapidi per finire in galera o per scrivere romanzi...

ESTRARRE LA MUSICA

□ In molti giochi commercializzati è presente una musica di sottofondo. Mi piacerebbe registrarla a parte per inserirla in miei listati (ed altre domande).
(Paolo Nannotti - Torrita)
(Giacomo Rizzo - Avellino)

• La musica che spesso è udibile in vari programmi non scaturisce da un'unica procedura software universale. C'è da dire, infatti, che di solito queste musichette girano all'interno del ciclo di interrupt; ciò significa, in altre parole, che sono in funzione due programmi contemporaneamente: il primo provvede a mantenere sempre attiva la musica, il secondo rappresenta il gioco vero e proprio.

Le tecniche di sonorizzazione sono pressochè infinite, non solo perchè la routine preposta può essere sistemata, dall'ideatore del gioco, in una parte qualunque della memoria, ma anche perchè il "dirottamento" dell'interrupt può avvenire in moltissimi modi; analogamente i codici che rappresentano le note musicali possono variare da gioco a gioco ed assumere significati diversi a seconda della procedura adottata. Tali inconvenienti rendono impossibile, in pratica, determinare una tecnica unica, valida per tutti i giochi in commercio (sprotezione a parte!).

Mi permetto di ricordare, tuttavia, che sul nostro dischetto mensile "Directory" sono presenti numerose musichette, spesso di notevole effetto, che puoi inserire senza problemi nei tuoi programmi dal momento che agiscono nel ciclo di interrupt: è

SOFTWARE

VETTORI E MATRICI

□ Che significato hanno i comandi del tipo...
DIM A$, X
...in cui non sono presenti gli argomenti? Inoltre vorrei divulgare una semplice tecnica per trasformare una matrice pluridimensionale in un vettore monodimensionale.
(Nunzio Santini - Comiso)

• Il dimensionamento di un vettore o di una matrice richiede esplicitamente gli argomenti, da racchiudere tra parentesi, solo nel caso in cui si voglia definire con molta precisione il numero di elementi che lo comporranno. In caso contrario viene automaticamente assegnato il valore di default, che è 10.

Dim A$, insomma, equivale a: Dim A$(10).

E' noto, comunque, che se ci si accontenta di 11 elementi per vettore (numerati da 0 a 10) è possibile fare a meno dell'operazione preventiva di dimensionamento. Anzi, è possibile (senza ricorrere a DIM) dimensionare IMPLICITAMENTE una matrice fino a tre dimensioni.

Per verificarlo, accendi il C/64 e digita, in modo diretto...

```
A (10, 10, 10) = 12
```

La risposta (Ready) confermerà che il dato è stato memorizzato. E ovvio che, altrettanto implicitamente, la memoria a disposizione diminuirà perchè il computer è stato costretto a far spazio per ospitare 11 x 11 x 11 elementi. Puoi verificare quanto affermato digitando Print Fre (0) prima e dopo l'operazione di assegnazione implicita.

Per ciò che riguarda la tecnica che hai inviato, relativa all'adattamento di una matrice in un vettore, l'argomento è molto interessante, ma è pevenuto su carta; ti consiglio, quindi, di inviarmelo nuovamente (corredato dei programmi dimostrativi) su supporto magnetico. Ho detto migliaia di volte che accettiamo solo articoli che pervengono su disco (in formato Easy Script o altri w/p).

sufficiente seguire le istruzioni allegate su ciascun dischetto.

Per ciò che riguarda il programmatore di Eprom (presentato sul n. 53 di C.C.C.) tieni presente che serve solo ai programmatori (per di più un po' esperti di l.m.). In una Eprom puoi inserire qualsiasi programma, anche i giochi commercializzati, a patto di esser capace di rimuovere la protezione e di effettuare i necessari, indispensabili, cambiamenti per adattare il software in zone di memoria probabilmente incompatibili con il gioco originale.

Se infine, come affermi, il nostro gioco "ZAGOR" è stato pirateggiato su larga scala, questa notizia da un lato ci fa piacere (perchè dimostra la buona qualità dei nostri prodotti); dall'altro ci amareggia perchè l'azione di pirateria viene condotta in un circuito commerciale, vale a dire per scopi di lucro, e non nel circuito, che spesso sosteniamo, dello scambio di programmi tra appassionati.

SOFTWARE

LEGGERE FILE

□ Vorrei modificare l'ottimo listato "Program - menu" (pubblicato sul n. 47) per poter leggere i vari file sequenziali registrati su un dischetto.
(Lauro Guglielmo - Caserta)

• Il programma cui fa riferimento il nostro lettore consente di leggere i nomi di tutti i file Prg presenti su floppy e di caricarne uno dopo averlo indicato con la semplice pressione di un tasto.

La modifica richiesta è sicuramente possibile. E' sufficiente, infatti, modificare la riga 230...

```
230 if (asc(a$) and 127) <>1 then 280
```

...e di sostituire l'intero blocco di righe, da 400 in poi, con quelle che seguono:

```
400 print: print: print,," file?"
410 get b$:if b$="" then 410
415 if b$< "a" or b$> "t" then 320
420 b=asc(b$)-65+l: if b<145 then print: print,,n$(b)
435 if b>144 then 320
436 if n$(b)="" then 320
440 print chr$(147);: open 1, 8, 8, n$(b)
450 get#1,a$: if st<>0 then 480
460 if a$="" then 450
465 get b$: if b$<>"" then 480
470 print a$;: goto 450
480 print: print: print"premi un tasto"
490 get a$: if a$="" then 490
500 close 1: goto 320
```

Con questi suggerimenti il programma, ovviamente, rimane sempre in memoria ed è sempre disponibile per esaminare il contenuto di altri file sequenziali. Naturalmente l'eventuale presenza, nei file, di caratteri speciali (come quelli relativi al movimento del cursore, cambio colore e così via) possono provocare "pasticci" sullo schermo. Basterà, in questi casi, premere Run/Stop e Restore per ripristinare le condizioni di default.

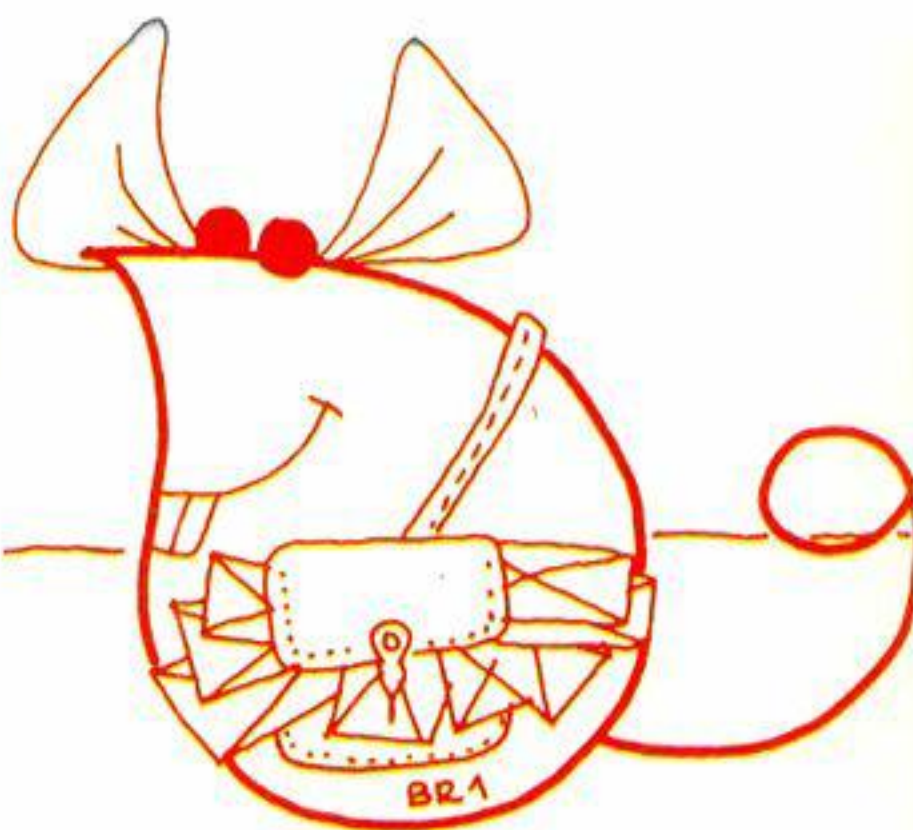
GRAFICA HI-RES

□ Come posso realizzare quei meravigliosi disegni che compaiono nei videogame?
(Simone Bellinazzo - Roverdicrè)

• Non esiste un programma specifico nè, tantomeno, istruzioni dedicate o "trucchi" più o meno conosciuti.

Per il C/64 sono disponibili numerosi Tool che consentono di trasformarlo in una tavolozza elettronica. E' come se si potesse disporre, invece di pennelli e tela, di joystick e schermo. I disegni vengono fuori esclusivamente dall'abilità del disegnatore e non da opportuni comandi grafici.

Tuttavia c'è da notare che, grazie ai computer (ed ai relativi, indispensabili programmi grafici) è possibile caricare immagini già memorizzate (oppure provenienti da un Tv, da una telecamera o da un videoregistratore) e modificarle a volontà; oppure effettuare mixage tra due immagini o altro ancora.

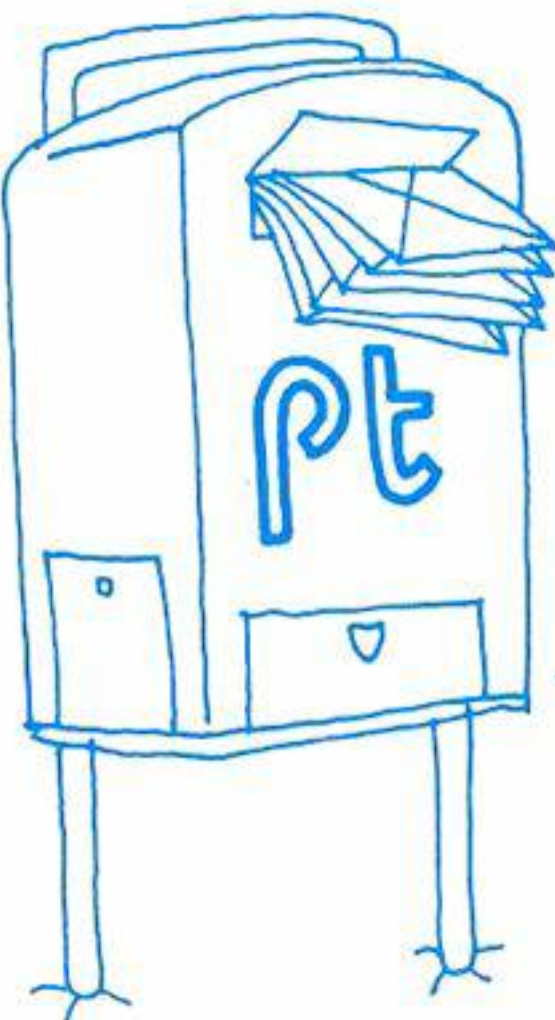
In ogni caso, però, il risultato dipenderà dall'estro (e dalla bravura) dell'utente.

□ E' vera la notizia secondo cui presto uscirebbe una scheda per trasformare il C/128 in Amiga?

Vorrei inoltre sapere se l'e-

mulatore C/64 per Amiga funziona anche con programmi l.m. Esiste, infine, un emulatore Ms-Dos per Amiga 500 e, in caso negativo, conviene acquistare l'Amiga 2000 dotato di scheda Ms-Dos?

(Roberto Corelli - Genova)

• Per ciò che riguarda la scheda Amiga per C/128, bè, non ho mai sentito una panzana più grossa; ciò dimostra che la vita è bella perchè è varia.

Per ciò che riguarda l'emulatore di C/64 per Amiga, la sua compatibilità è a volte stupefacente. Ho visto girare perfino programmi l.m. (per C/64) che ricorrevano a manipolazioni dell'interrupt. E' bene sottolineare, tuttavia, che molti programmi non girano, oppure girano parzialmente (cioè all'improvviso si bloccano) oppure girano a velocità più ridotta che sul C/64.

Forse è solo questione di software; speriamo che, in futuro, siano disponibili emulatori sempre più sofisticati.

Per ciò che riguarda la scheda Ms-Dos per Amiga, attualmente è disponibile solo per il modello A-2000. Tocca a te decidere se, a parità di spesa (o quasi) è meglio un A-2000 dotato di scheda Ms-Dos oppure due computer: un A-500 ed un IBM compatibile.

FLOPPERIA ® srl

Viale Monte Nero, 31
20135 Milano
Tel. (02) 55.18.04.84

Vendita per corrispondenza in tutta Italia
Evasione ordini in 24 ore
Assistenza hardware/software, riparazioni e consulenza

HARDWARE

Amiga 500/2000....telefonare
Amiga 2500....telefonare
Janus XT con drive 5" 1/4....990.000
Janus AT....1.750.000
Scheda velocizzatrice con i nuovi processori 68020-68881 ed eventuale compatibilità Unix....telefonare
Espansioni per tutti gli Amiga....telefonare
Monitor 14" Philips 8833 stereo....550.000
Monitor 14" A2080 alta persistenza....690.000

DISCHETTI DS/DD

3" 1/2 bulk....2.300
3" 1/2 GMC....2.800
3" 1/2 Nashua....3.000
5" 1/4 bulk....1.000
5" 1/4 bulk colorati....1.500
5" 1/4 GMC....2.000
5" 1/4 Nashua....2.500
5" 1/4 GMC hd....5.000
minimo 10 pz. - sconti per quantità

PREZZI IVA 19% INCLUSA

I prezzi potranno variare a seconda dell'andamento delle valute estere

HARD DISK

HD 20 MB....599.000
HD 40 MB....890.000
HD 40 MB, accesso in 40 ms....999.000
HD 40 MB velocissimo - 25 ms....1.290.000
HD 70 MB, voice coil, 30 ms....2.350.000
tutti gli hard disk sono completi di controller per IBM o Janus XT/AT
HD 20 MB per Amiga 500....999.000
HD 20 MB per 2000 in AmigaDOS....1.090.000

MINIRACK

Mobiletto per Amiga 500 o per Atari ST, sostiene il monitor e fornisce spazio per contenere drive, alimentatore, cavi, riviste, ecc.
£. 69.000

SUPPORTI MONITOR

Robustissima base rotante su 360 gradi, inclinabile di 25 gradi, con piedini antivibrazione ed antislittamento.
9-13 pollici.... 35.000
14-18 pollici.... 40.000

SUPPORTO TOWER

Sistema l'Amiga 2000 in verticale sul pavimento, per risparmiare spazio sulla scrivania e dare un tocco di professionalità al vostro sistema.
£. 59.000

CASSETTO A SCOMPARSA

Per inserire la tastiera dell'Amiga 2000 sotto il computer, come nel modello 1000, e recuperare spazio sulla scrivania.
£. 89.000

JITTER-RID

Filtro antiriflesso per monitor, riduce lo sfarfallio ed aumenta contrasto e definizione.
12" mono.... 35.000
14" color.... 40.000

LEGGII

Per facilitare la battitura di lettere e di listati, anche da riviste. Disponibile in 3 versioni a partire da
£. 19.000

DUST REMOVER

Maneggevole mini-aspirapolvere er rimuovere la polvere che si accumula in tastiere, schede, ecc.
£. 25.000

DUST COVER

Copertina trasparente antistatica, protegge da polvere e liquidi dannosi.
per A-2000.... 18.000
per stampanti 80 col.... 15.000
per stampanti 132 col.... 18.000

Disponibile l'intera libreria di software Public Domain di Fred Fish
Richiedeteci il catalogo su disco che vi sarà spedito in contrassegno di £. 10.000

DRIVES

Drive 3" 1/2 Amiga, slim line, compatibile 100%, con connettore passante; in **omaggio** disco utility.
£. 239.000
Drive 3" 1/2 interno per Amiga 2000, colore beige, 100% compatibile; in **omaggio** disco utility.
£. 199.000

1.8 MEGABYTE

Espansione per Amiga 500, porta la memoria totale del computer a 2.3 MegaByte. Si inserisce nello slot previsto per le espansioni da 512k sotto la tastiera, autoconfigurante, con interruttore per il disinserimento, non invalida la garanzia.
Telefonare

VIDEO

EasyView: digitalizzatore video, operante in tutti i modi grafici, con filtri e software....130.000
Videon: digitalizzatore video in standard PAL, permette di digitalizzare a colori direttamente da qualsiasi sorgente senza l'uso di filtri esterni....telefonare
VD-Amiga digitalizzatore in tempo reale di ottima qualità video (framegrabber).... telefonare
VideoMaster nuovo mixer video Commodore.... telefonare
Telecamera b/n alta definizione....399.000
Genlock amatoriali e professionali....da 400.000

AUDIO

EasySound: campionatore audio con microfono e software standard IFF....130.000
Interfaccia MIDI standard, professionale ed espandibile....59.000

STAMPANTI

Commodore MPS 1250....490.000
Commodore MPS 1500 a colori....590.000
Star LC-10, 140 cps, 80 col., bidirez., NLQ....520.000
Star LC-10 versione a colori....620.000
Nec P-2200, 170 cps, 80 col., 24 aghi, bidirez., con 5 fonts NLQ residenti....950.000

PORTASTAMPANTI

Disegno funzionale, robusta costruzione in metallo, con supporto angolato per consentire la lettura durante la stampa.
80 col.... 29.000
132 col.... 39.000

ACCESSORI

Copritastiera per Amiga 500....19.000
Kit pulizia testine drive 3" 1/2....10.000
Kit pulizia testine drive 5" 1/4....10.000
Portadischetti 3" 1/2 40 pz....20.000
Portadischetti 3" 1/2 60 pz....25.000
Portadischetti 5" 1/4 40 pz.... 20.500
Portadischetti 5" 1/4 100 pz....30.000
Portadischetti 3" 1/2 150 pz. Posso....35.000
Portadischetti 5" 1/4 150 pz. Posso....42.000
Mobili portacomputer....da 99.000

64 EMULATOR

La nuova versione del famoso emulatore C64, con gestione dell'audio, sprite, stampanti e drive dedicati; utilizza i drives Amiga, hard disk compresi.
£. 29.000

ROM KICKSTART 1.3

£. 299.000

risposte rapide

SPRITE PER C/16
(Nicola Spoto - Casteltermini)

Il programma che hai inviato su disco, e che aggiunge nuovi comandi al C/16 per creare e gestire sprite, è molto interessante ma, come riconosci tu stesso, è troppo lungo e complesso da digitare (e quindi da pubblicare). Se mi autorizzi, lo inserirò in uno dei prossimi dischetti "Directory", come pure il simpatico "Scacchi" anch'esso in l.m. Se decidi in tal senso, inviaci altri listati dimostrativi (sempre su disco) per la gestione degli sprite. Perchè, comunque, continui a dedicare energie ad un computer che, ormai, non ha più niente da dire? Non sarebbe meglio un bel C/64 (per non parlare di Ms-Dos e Amiga)...

ISTRUZIONI INSUFFICIENTI
(Gaetano Piazza - Francofonte)

Il programma che hai inviato è privo di un demo; d'altra parte, nell'articolo accluso, non hai indicato nemmeno un esempio pratico su come gestire i numerosissimi parametri che entrano in gioco. Di conseguenza il listato è forse bello ma, così come è, rimane inutilizzabile.

SCHERMO DI SCORTA
(Alfonso Daniele - Alghero)

Mi fa piacere che, grazie all'articolo pubblicato sul n. 54, tu sia stato in grado di apportare l'interessante miglioria che descrivi nella lettera. Purtroppo, per sapere se merita di esser pubblicata, è indispensabile inviare programma ed articolo su disco; in caso contrario i lavori non possono essere accettati.

GIA' PUBBLICATI

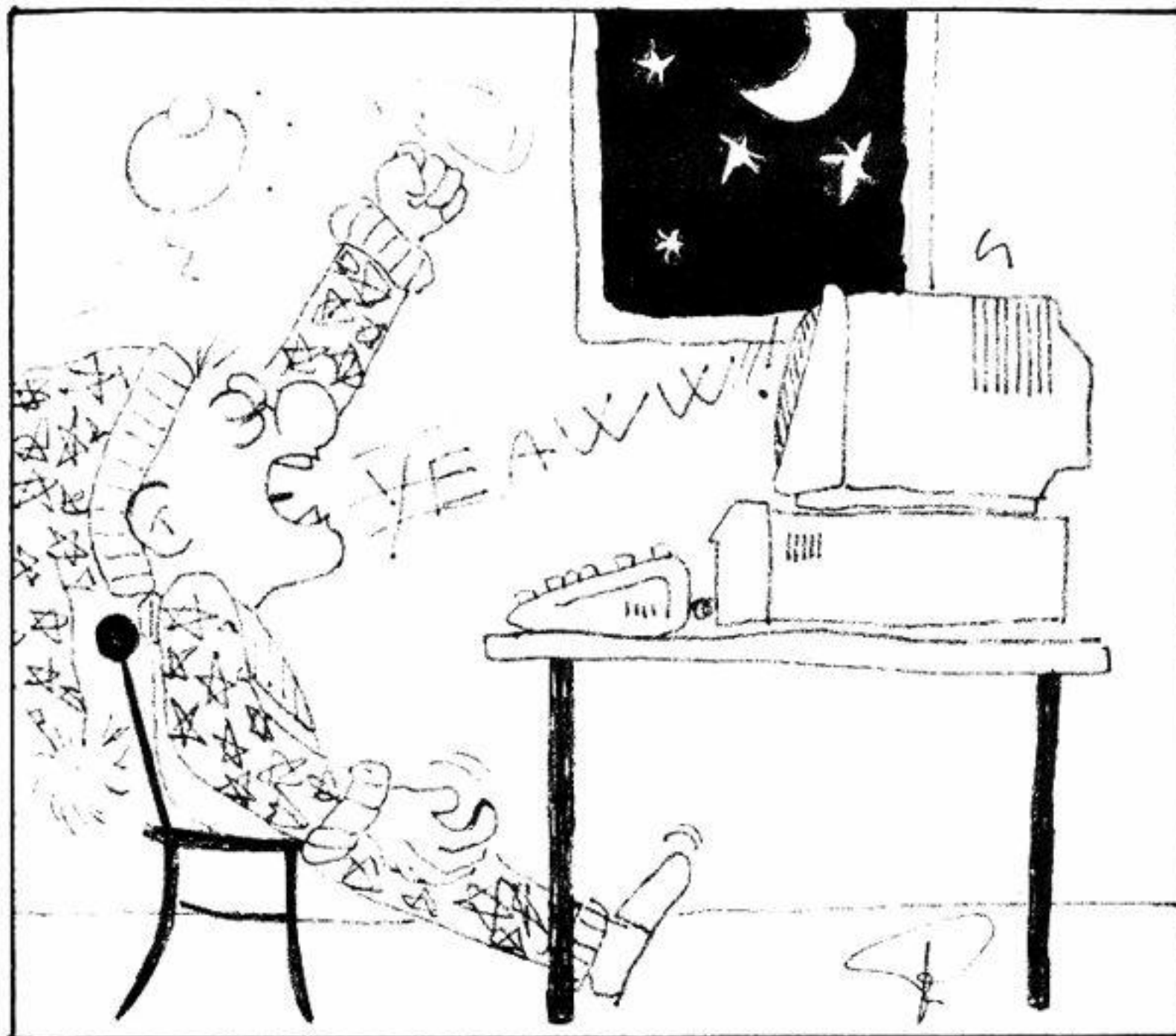
(Massimiliano Morando - California)
Sul N. 51 è stato pubblicato un intero articolo (e relativi programmi) che spiega in modo dettagliato la gestione degli sprite con il C/128.

(Mario Morniroli - A. di Taggia)
Una descrizione sull'hard disk puoi "estrarla" dall'articolo "Monologo sopra i massimi sistemi", pubblicato sul n.55.

VUOI COLLABORARE CON COMMODORE COMPUTER CLUB?

Molti lettori desiderano proporre articoli e programmi da pubblicare sulla nostra rivista. Per consentire l'esame del materiale proposto, ma anche per evitare spiacevoli malintesi, riportiamo qui di seguito le norme da seguire per veder pubblicato (e compensato!) un proprio lavoro.

- NON inviare programmi e/o articoli che non siano stati espressamente concordati con la Redazione.
- Per risolvere il problema precedente, telefonare in Redazione (02 / 84.67.34.8) per proporre la collaborazione.
- Si tenga ben presente che non potranno essere accettati programmi che, pur rispondenti al tema concordato, siano privi di corrispondente articolo esplicativo.
- Una volta concordato il "tema", e portato a termine articolo ed eventuale programma, inviarli esclusivamente su FLOPPY DISK; tale necessità deriva dal fatto che, per questioni di tempo, non ci è possibile digitare nè l'uno nè l'altro. E' facoltativo l'invio del lavoro riprodotto anche su carta.
- Per redigere l'articolo, servirsi di un Word Processor commercializzato: Easy Script C/64, Superscript C/128, Word Pro 3, Wordcraft Vic 20, W/P del Plus/4, Magic Desk o altri concordati nel corso della suddetta telefonata. Nel caso di Amiga usare solo file di "Notepad" ed evitare dischetti in auto-boot. Sono da escludere W/P ideati e scritti dall'autore, anche se allegati sul disco inviato.
- I lavori verranno compensati solo ad avvenuta pubblicazione ed in proporzione alla qualità, interesse e rispondenza a quanto concordato; la cifra minima è fissata in L. 25000 per pagina pubblicata. Esempio: articolo e programma che occupano 4 pagine della rivista: compenso minimo L. 100 mila.
- Gli articoli che non dovessero esser pubblicati (a causa della mancata rispondenza a quanto concordato, o per altri motivi) non verranno compensati.
- La Systems Editoriale si riserva il diritto di riportare il materiale inviato, in tutto o in parte, sulle altre testate.
- I lavori inviati allo scopo di comparire soltanto su "Directory", oppure "Amigazzetta", saranno compensati con materiale prodotto dalla Systems Editoriale che si impegna, altresì, a mantenere in evidenza il nome dell'autore nei file inviati.
- Il materiale inviato non viene restituito al mittente in nessun caso.
- Gli articoli e/o programmi inviati sono da considerare a totale disposizione della Systems Editoriale, fatte salve le norme precedentemente esposte.

Rispettando le disposizioni di cui sopra eviterete perdite di tempo, di denaro (per spedizioni non richieste) e, soprattutto, di pazienza.

ALLA SCOPERTA DI "DIRECTORY"

Ecco quali sono i file che troverete periodicamente sul disco in vendita solo per corrispondenza

Nel lontano ottobre 1986, considerato lo straordinario successo di vendita dei disk drive 1541, la Systems Editoriale prese la decisione di offrire, ai propri lettori, la possibilità di procurarsi, su disco, i programmi pubblicati su "Commodore Computer Club".

Il primo numero di "Directory" (questo fu il nome attribuito alla fortunata collana) conteneva tutti i programmi comparsi sul n. 34 di Commodore Computer Club (C.C.C.), tutte le routine dell'enciclopedia Basic fino a quel momento pubblicate, vari giochi ed altri programmi e file di interesse generale.

I blocchi lasciati liberi su disco risultavano solo 34; ciò significava una raccolta soft di oltre 160 mila byte! L'abitudine di riempire quasi per intero il dischetto fu mantenuta anche nei numeri successivi.

In seguito, sempre su Directory, furono inserite musiche nell'interrupt, schermate grafiche, set di caratteri ridefiniti ed altri file e programmi di notevole interesse.

Da notare che NESSUN programma risulta protetto ed i file in l.m. possono essere agevolmente esaminati con un qualsiasi Monitor per linguaggio macchina.

Questo conferma che la Systems Editoriale, prima tra le case editrici, promuove lo scambio (SENZA FINI DI LUCRO) tra appassionati che, grazie ad una spesa irrisoria, possono procurarsi software da utilizzare immediatamente, oppure dopo aver apportato modifiche e personalizzazioni.

Il prezzo al pubblico, infatti, è fissato in L. 12000 per ciascun dischetto (oltre alle spese di imballo e spedizione).

Recentemente molti nuovi lettori ci hanno chiesto di indicare l'abbinamento tra numero di Directory e numero del fascicolo di C.C.C. i cui programmi sono registrati.

Riportiamo, pertanto, le delucidazioni richieste, precisando che, oltre ai programmi pubblicati su C.C.C. sono riportati, su CIASCUN dischetto Directory, moltissimi altri programmi, a volte inediti, a volte tratti da altre pubblicazioni della Systems Editoriale.

Ma ecco la corrispondenza tra i 19 numeri di Directory finora usciti e quelli di C.C.C.:

1: Enciclopedia di routine Basic + C.C.C. n. 34
2 :C.C.C. n. 35 + n. 36
3 :C.C.C. n. 37

4 :C.C.C. n. 38
5 :C.C.C. n. 39
6 :C.C.C. n. 40
7 :C.C.C. n. 41
8 :C.C.C. n. 42
9 :C.C.C. n. 43
10 :C.C.C. n. 44
11 :C.C.C. n. 45
12 :C.C.C. n. 46
13 :C.C.C. n. 47
14 :C.C.C. n. 48
15 :C.C.C. n. 49
16/17 :C.C.C. n. 50 + C.C.C. n. 51
18 :C.C.C. n. 52
19 :C.C.C. n. 53
20 :C.C.C. n. 54 + C.C.C. n. 55
21 :C.C.C. n. 56 + C.C.C. n. 57

SUDDIVISIONE DI DIRECTORY

I nuovi numeri di Directory saranno strutturati in modo più razionale per consentire, anche ai lettori meno esperti, di utilizzare in pieno le notevoli risorse del software ivi memorizzato.

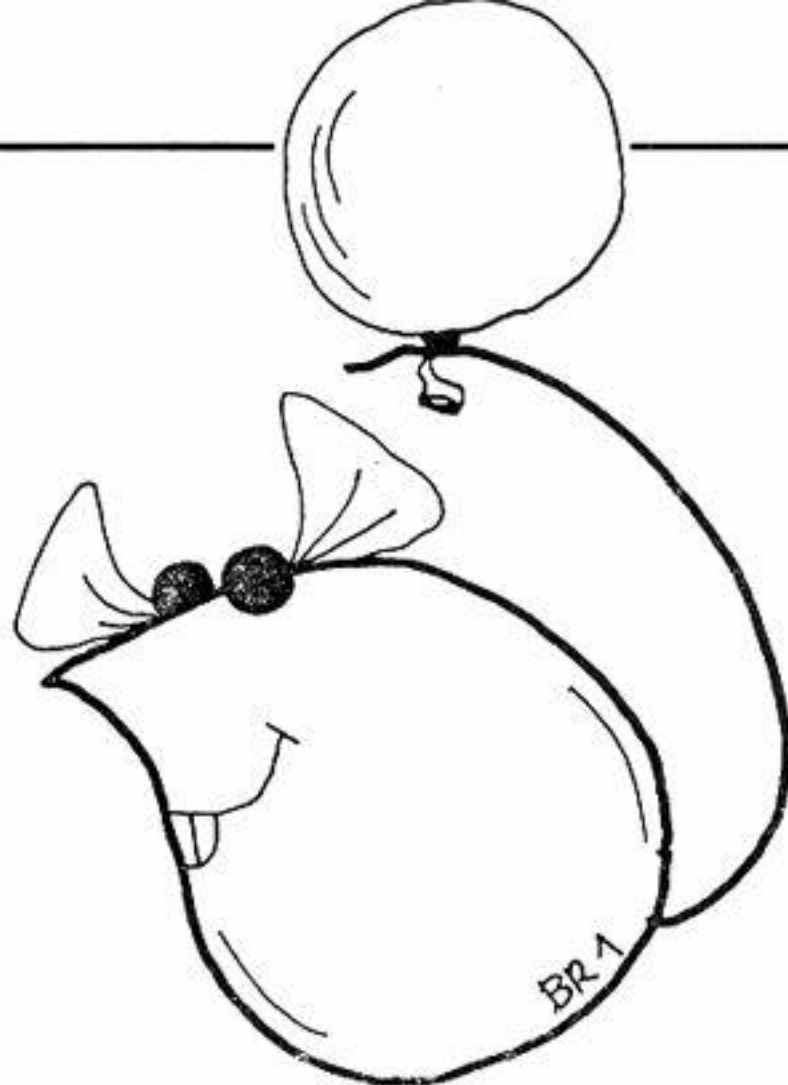
Il primo programma del floppy avrà sempre il nome "caricami" e, una volta lanciato, fornirà le informazioni di massima per il corretto uso del dischetto.

Ciascuno di questi, infatti, è normalmente diviso in cinque sezioni che ora esamineremo una per una.

C.C.C. N. XX

In questa sezione verranno inseriti tutti i programmi pubblicati su C.C.C. il cui numero, come intuitivo, è rappresentato da XX. Sono qui presenti eventuali file di "supporto" (schermate grafiche ed archivi di esempio, se necessari al buon funzionamento dei vari programmi) come pure eventuali versioni migliorate dei programmi che appaiono sul fascicolo di C.C.C. citato.

SYSTEMS EXTRA

Appartengono a questa sezione i programmi eventualmente tratti da altre pubblicazioni della Systems Editoriale concernenti, ovviamente, l'ambiente Commodore. Tra le pubblicazioni ricordiamo "Personal Computer", "Software Club" e "Commodore 64 Club".

OLD C.C.C.

In questa sezione i lettori troveranno una "ristampa" di programmi già pubblicati su C.C.C. in precedenza e che, per vari motivi (tra cui le richieste dei lettori), si ritiene siano di interesse generale.

EXTRA FILES

E' questa la sezione in cui vengono riportati programmi di vario tipo: musiche nell'interrupt, schermate grafiche, software di dominio pubblico, archivi di vario tipo, giochi, eccetera.

SPAZIO LETTORI

In questa sezione, infine, verranno ospitati i lavori inviati dai lettori tra cui programmi troppo lunghi (o troppo particolari) per esser pubblicati sulla rivista, migliorie di programmi già apparsi, personalizzazioni di software di dominio pubblico ed altre... curiosità che i lettori vorranno inviare.

E allora, buon software con "Directory" della Systems Editoriale!

COME PROCURARSI "DIRECTORY"

Avvertiamo i lettori che NON ci è assolutamente possibile inviare i programmi su nastro, per intuibili motivi di economia ed affidabilità del nastro cassetta.

"Directory" può quindi esser richiesto solo su disco, inviando L. 12000 per ciascun floppy desiderato, oltre alle spese di confezione, imballo e spedizione (L. 3000 per spedizione "normale"; L. 6000 per inoltro a mezzo raccomandata)

Si sottolinea che NON ci è possibile inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di indirizzo (ed eventuale telefono), ma anche il titolo del prodotto desiderato; esempio:

Mario Rossi
Via dei giardini 12
00123 Roma
(Tel. 06/12.34.56)

Desidero ricevere:
"Directory" N.1, N.7, N.12

La somma risulta:
Dischetti L. 12000 x 3 = L. 36000
Spediz. raccom. L. 6000
Totale: L. 42000

E' intuitivo che, con lo stesso ordinativo, potete chiedere altri prodotti della Systems Editoriale (arretrati, libri, dischi e così via) il cui prezzo di vendita è indicato nelle apposite pagine informative presenti anche in questo stesso fascicolo.

Si consiglia, per coloro che dispongono di un conto corrente presso QUALSIASI banca o Istituto di Credito, di inviare la somma dovuta non a mezzo c/c postale, ma a mezzo assegno (non trasferibile o sbarrato due volte) da inviare, in busta chiusa, al nostro indirizzo, corredandolo di lettera di accompagnamento.

In questo modo la spedizione dei prodotti richiesti sarà evasa in tempi notevolmente più brevi.

Un package professionale per disegnare, giocare e produrre immagini extracolorate con Amiga

A.A.A(miga): DISEGNATORE ESPERTISSIMO OFFRESI

di **Luigi Callegari**

The Butcher (versione 2.0) è un sofisticato package per la manipolazione di immagini grafiche su qualunque computer Amiga (2000, 1000 e perfino A-500 dotato di memoria standard). E' uno strumento grafico "complementare", di ausilio per gli utenti di Deluxe Paint, Digipaint e programmi del genere. Risulta inoltre praticamente indispensabile per gli utilizzatori di immagini grafiche digitalizzate da videoregistratore.

MODI E RISOLUZIONI

The Butcher usa lo standard PAL, dunque le risoluzioni in pixel sono: 320 x 256, 320 x 512, 640 x 256 e 640 x 512. Il programma può caricare anche file in formato HAM, ma prevede la possibilità di manipolare solo schermate in formato standard; è quindi necessario dapprima effettuare una conversione, usando le apposite opzioni da menu (HAM to LO-RES o HAM to HI-RES). Si noti, tuttavia, che per convertire un file HAM in un file HI-RES a colori è necessario disporre di almeno un MegaRAM di memoria.

BITPLANES

Ripetiamo, Butcher non è un programma per creare disegni, sebbene comprenda utility di tracciatura rudimentali (menu TOOLS opzione DRAW), ma per lavorare su disegni già fatti. Esso permette di manipolare in una maniera molto profonda lo hardware della macchina, quindi per usarlo al meglio è necessario conoscere bene come viene generata la grafica da Amiga ed aver letto, ad esempio, il nostro articolo a pagina 33 del numero 49 di CCC, dove spieghiamo diffusamente che cosa significa Bitplanes e come interagiscono con Denise (il chip grafico di Amiga).

IL MENU "PROJECT"

Comprende 16 opzioni principali, essenzialmente per caricare (OPEN) un file da disco o salvarlo con nome prefissato (SAVE) o da specificare (SAVE AS). E' possibile anche cancellare un file (DELETE) e chiudere il Workbench per risparmiare memoria (CLOSE WORKBENCH). Si noti che nella finestra di gestione del caricamento e salvataggio di schermi compare un rettangolino con la scritta "Make Icons" commutabile in "No Icons", clickandoci. Nel primo caso, quando il computer salva l'immagine su disco, salva anche una icona, sotto forma di un file del tipo "nome.info" che accompagna il file "nome" e che ne permette la comoda gestione "sotto" Workbench.

Le opzioni PRINT CLIP e PRINT permettono di stampare su periferica grafica la videata. La stampante deve essere stata configurata da Preferences, come spiegato sul manuale fornito con Amiga. PRINT CLIP stampa solo una CLIP, cioè una porzione di schermo definita via mouse (come una Brush in Deluxe Paint). Molte opzioni di Butcher, infatti, permettono di lavorare solo su di una porzione ristretta e definita del video, come se il resto non esistesse.
SPARE PICTURE prevede diversi menu per manipolare una porzione di memoria di Amiga che viene usata per contenere un altro schermo non visualizzato. Questo schermo nascosto può essere scambiato (SWITCH), cancellato (KILL), miscelato in vari modi (MIX in FRONT / BACK). COPY TO SPARE salva in tale porzione di memoria lo schermo attualmente visualizzato.

Tramite l'opzione FORMAT è possibile definire la dimensione dello schermo ed il numero di bitplanes (quindi, il numero massimo di colori consentito). Tramite icone è possibile fissare il modo ad alta risoluzione (HIRES) ed interlacciato (LACE) e Hold - and - Modify (a 4096 colori).

L'opzione OVERSCAN permette di disegnare senza bordo schermo.

L'opzione DELETE UNDO MAP permette di rilasciare la zona di memoria usata dal programma per memorizzare lo schermo attuale prima di eseguire una operazione. Ciò permette di annullare (UNDO) l'ultima

AMIGA, UNA GRAFICA STUPENDA

Dal momento che le prestazioni grafiche di Amiga costituiscono, tuttora, il top tra i personal computer di basso costo (e non solo tra questi), le numerose software house che producono programmi per Amiga hanno divulgato una quantità notevole di titoli.

A parte i programmi "storici" per creare animazioni (Aegis Animator), magari tridimensionali (Videoscape) ed in ray-tracing (Sculpt 3D), i desktop (Shakespeare), i CAD (PRO-CAD) più o meno specifici (PCLO per circuiti stampati), esistono programmi di disegno classico, tipo Deluxe Paint e Digipaint che devono la loro spettacolarità ed il loro successo semplicemente ad un (ancora ristretto) sfruttamento dello hardware superiore di Amiga.

THE BUTCHER (il macellaio, letteralmente), è un programma assai originale, che si discosta notevolmente da tutti gli altri presenti sul mercato e non solo per Amiga. Il suo uso è abbastanza intuitivo, grazie al mouse ed ai menu, ma crediamo che tutti potranno imparare qualcosa da queste note, specie chi non possiede il manuale o non conosce l'inglese o non ha il tempo di leggerlo.

operazione semplicemente pigiando i tasti AMIGA-U (o selezionando da menu TOOLS - Undo). Cancellando questa zona il programma disporrà di più memoria.

Altra opzione che consente di risparmiare memoria è la PROGRAM LOADING: scegliendo PARTIAL, il programma mantiene residente (contrariamente alla norma) solo il programma centrale e carica quindi, di volta in volta da disco, le porzioni di programma che servono a svolgere una funzione richiesta. Come Workbench ed i comandi del CLI, insomma.

MENU "TOOLS"

Le opzioni di questo menu servono essenzialmente per manipolare grossolanamente lo schermo.

Le opzioni REVERSE e FLIP permettono di ruotare lo schermo ortogonalmente.

L'opzione DRAW accede ad un sotto-menu di tracciatura di linee, cerchi, spezzate molto elementare.

CYCLE fa ruotare eventuali sequenze di colori prefissate (come premendo TAB sotto Deluxe Paint), permettendo le stesse banali animazioni grafiche.

MAKE PALETTE permette di definire una paletta di colori. Compare un menu di definizione completa di slider (regolatori lineari grafici) che definiscono la quantità di Rosso, Giallo e Blu per un dato colore; oppure è possibile regolare la saturazione e la brillantezza, che alterano automaticamente le varie percentuali di colore.

Inoltre, caratteristica esclusiva del programma, è possibile regolare i colori fondamentali (rosso, giallo e blu) o saturazione e brillantezza di tutto il disegno contemporaneamente. Come quando si stampa una foto da un negativo a colori, è così possibile regolare perfettamente le eventuali "dominanti" cromatiche di una foto digitalizzata, oppure ritoccare creativamente un disegno già fatto, virando i colori in blocco per dare, ad esempio, una tinta dominante.

Il menu di creazione della palette permette anche di complementare, scambiare, riordinare i colori nella paletta, che sul video vengono presentati come blocchetti ordinati, corrispondenti ai vari registri hardware di Denise. Un colore può essere scelto tra i blocchetti visualizzati,

RISOLUZIONI E FORMATI

Per chi si fosse fatto sfuggire l'articolo sul numero 49 di CCC, ricordiamo molto brevemente alcune note fondamentali sui possibili formati grafici di Amiga:

- 320 x 256 o 640 x 256 con al massimo 32 colori (6 bitplanes) scelti senza limitazioni da una paletta di 4096 tinte.
- 320 x 512 o 320 x 512 con al massimo 16 colori (5 bitplanes) scelti senza limiti da una paletta di 4096 tinte. L'immagine video sfarfalla se non si usa un monitor ad alta persistenza.
- 320 x 256 o 320 x 512 fino a 4096 colori contemporaneamente a video in modo HAM, ma con alcune limitazioni sui possibili colori adiacenti orizzontalmente.

Il computer è in grado di presentare sul video immagini con palette di colori e risoluzioni differenti (non sulla stessa linea, tipicamente) direttamente da sistema operativo, grazie ad Intuition.

I file (grafici, sonori e di testo), caratteristica utilissima ed unica di Amiga, usano un formato ormai standardizzato detto IFF (Interchange File Format) messo a punto dalla Electronic Arts ed adottato dalla Commodore. Quindi, un file generato da un programma può essere trattato anche da qualunque altro package senza bisogno di conversioni.

oppure preso (PICK) dallo schermo. Butcher memorizza internamente (caratteristica stranamente assente in Deluxe Paint) più palette di colori, che possono essere scambiate tra loro (indicate con P1, P2 e P3 sul menu di creazione).

MENU "EFFECT"

E' senz'altro il menu più creativo di Butcher, in quanto, come suggerisce il nome, contiene varie opzioni per generare effetti insoliti sul disegno. Il loro uso, comunque, si comprende non tanto descrivendole, ma provandole, magari aiutandosi, subito dopo una operazione, con l'UNDO.

TONER permette di ottenere, tramite la regolazione di due slider grafici (cursori) una immagine ad un colore, più o meno saturo, tra numerosissime sfumature, come quando si fotocopia una immagine con un fotoriproduttore ad inchiostro colorato.

POS-NEG commuta l'immagine tra negativo e positivo, come se fosse, appunto, un negativo fotografico.

COMPLEMENT è simile al precedente, ma più adatto per immagini a colori.

FALSE COLORS sostituisce i colori con altri falsati, ad esempio come nelle cartine geografiche delle previsioni del tempo, in base alle loro quantità.

PSEUDO COLORS è simile al precedente, ma il suo effetto è irreversibile (se non usando l'opzione UNDO), in quanto i registri di Denise, cioè la paletta attuale di colori viene modificata effettivamente in funzione delle attuali concentrazioni di colori in modo non "simmetrico".

BLACK & WHITE trasforma un'immagine a colori nella corrispondente immagine in bianco e nero.

ANTIQUE trasforma l'immagine, solitamente già trasformata in bianco e nero, in una foto tipo "daguerrotype", cioè tipo antica foto da lastra, inserendo una dominante color seppia / marrone.

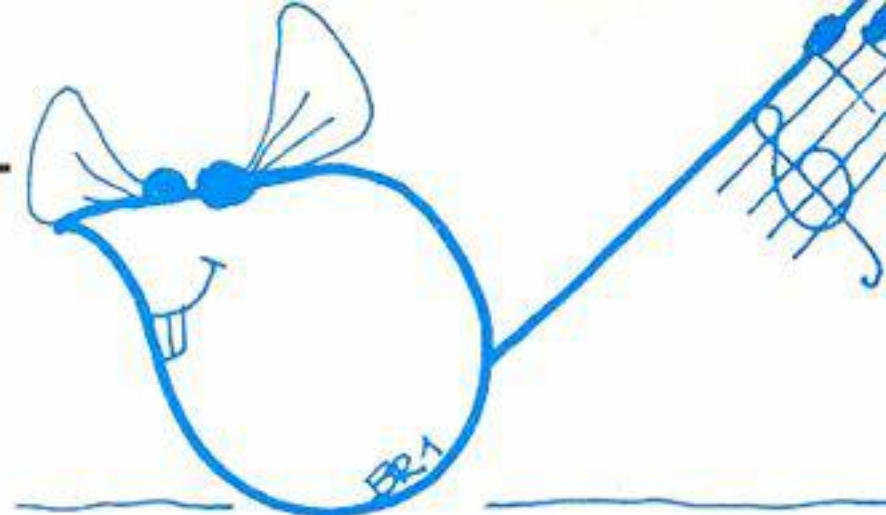
SEPARATE permette di ottenere una immagine in bianco e nero in base ad un solo colore tra sei. Ciò permette di effettuare, con più passaggi, una immagine da montare poi tipograficamente (se si dispone di una stampane grafica e di fogli lucidi) con processo addittivo o sottrativo.

BAND permette di isolare nel disegno i tre colori fondamentali (Rosso, Verde, Blu).

MENU "PROCESS"

Comprende opzioni per manipolare il disegno in modo "scientifico" piuttosto che creativo.

DGE è una opzione che attiva una finestra di controllo e permette operazioni grafiche notevoli. Ad esempio, è possibile "rafforzare" i limiti di separazione tra colori, per ottenere immagini più nitide e contrastate (come con un "intensificatore di immagini"). Oppure è possibile "rimappare" completamente un disegno in base a colori di sfondo differenti.

FILTER è una opzione tra le più potenti del programma, particolarmente utile per lavorare su immagini digitalizzate da telecamera. E' possibile ripulire dal programma i blocchi di pixel di un colore definito isolati, che spesso sporcano le immagini ottenute da telecamera, in base a filtri programmabili di varie intensità (più è alto il numero di filtro, più pixel vengono tolti, anche se ravvicinati). Ovviamente il sistema può essere usato anche per addolcire immagini eseguite dall'utente con disegno a "spruzzo" da Deluxe Paint oppure immagini convertire, con lo stesso Butcher, da HAM ad IFF.

MOSAIC effettua un disegno a mosaico. In pratica, grossomodo, trasforma il disegno attuale in uno con una risoluzione grafica di un quarto circa.

SHAPE MOSAIC permette di definire la sagoma con cui viene effettuata la creazione del mosaico, tipicamente un quadrato.

SLICE PLANE permette di "tagliare fuori" ovvero di lasciare attivi solo i Bitplanes che interessano. In questo modo è possibile anche controllare la riduzione, senza grosse perdite, del numero di Bitplanes necessari per rendere il disegno evitando di perdere troppi colori; oppure per vedere come sono distribuiti, in memoria, i vari colori.

DENSITY SLICE permette di visualizzare con un istogramma verticale le percentuali di colori usate per ogni registro di Denise (colore della paletta) e di inserire o disinserire i vari colori anche sinngolarmente, rendendo possibile isolare e lavorare su alcuni di essi oppure verificare visivamente quali colori influiscono di più sul disegno, ed in quali punti.

DIFFUSION trasforma il disegno in uno a due colori con le densità dei colori trasformati in sequenze di punti (come quando si usa una stampante grafica a due colori).

COLORI, CHE PASSIONE

In Butcher, come in molti altri programmi per Amiga (ed anche per altri computer) i colori vengono controllati in due modi, detti RGB ed HSL. Con il primo sistema si regolano (solitamente tramite potenziometri o valori numerici da uno a dodici in Amiga) la quantità di colore Rosso, Verde e Blu per creare una particolare sfumatura. I tre colori, detti fondamentali, permettono di ottenere tutti gli altri dello spettro ottico, semplicemente miscelandoli tra loro. Il sistema HSL (Hue, Saturation and Luminance), invece, dosa la Tinta, Saturazione e Luminosità per ottenere una sfumatura, con un sistema similè ai vecchi TV color. In ambedue i casi, sotto Butcher (dove spesso coesistono, ad esempio nei menu di Palette) si possono ottenere tutte le 4096 sfumature Amiga.

MENU "CONVERT"

Questo menu permette numerose "conversioni" intelligenti di immagini.
HISTOGRAM attiva una finestra di opzioni che permette di manipolare direttamente i registri di Denise, cioè la paletta di colori usando un istrogramma. E' possibile persino contare (CNT) quanti pixel ci sono di un certo colore e riordinare (SRT) la paletta in modo che i colori più usati vengano assegnati ai primi registri di Denise o agli ultimi. Inoltre, è possibile miscelare (MRG) due o più colori e, come sempre, lavorare solo su di una CLIP (CLP), cioè su una finestra di schermo definita col mouse.
QUICK MERGE attiva una finestra di menu praticamente identico alla precedente, ma lavora più velocemente in quanto non vengono scambiati i Bitplanes per risparmiare memoria.
AUTO MERGE permette, tramite finestra di opzioni, di scambiare le palette di colori (ricordiamo che sono tre in Butcher) direttamente. Eventualmente, è possibile anche usare un numero diverso di colori nella conversione tra palette. In pratica diviene così possibile fare eseguire automaticamente ciò che le precedenti due opzioni consentono di controllare manualmente.
CONSOLIDATE effettua un riordino automatico dei colori, in modo che vengano ordinatamente assegnati i più usati ai primi registri di Denise e scambiando poi adeguatamente i BitPlanes, in modo da risparmiare memoria. Effettuare un HISTOGRAM - Cnt subito dopo per visualizzare la nuova distribuzione dei colori!
SANITIZE è una opzione praticamente identica alla precedente, ma vengono soppresse eventuali doppie presenze di colori identici.
FLUSH ZERO riempie col colore di sfondo i pixel inutilizzati nel disegno.
CHANGE DEPTH permette di modificare brutalmente la profondità del disegno, ovvero il numero di Bitplanes (quindi, di colori) usati. La tavolozza di colori viene automaticamente ricalcolata per minimizzare le perdite di colori e garantire comunque la massima qualità.

Tutte le altre opzioni del menu permettono di effettuare, intuitivamente, conversioni tra formato standard, interlacciato e HAM nei due sensi. Ovviamente, talune di queste conversioni, dovendo mantenere due schermi contemporanemanete in RAM (uno originale e l'altro in creazione) richiedono spesso un Mega di memoria o più.

L'ESPERIENZA INSEGNA

Molti usano programmi come Butcher per realizzare dischetti dimostrativi di immagini, collezionandole tra le centinaia in circolazione per Amiga e personalizzandole o aggiungendone di proprie. Si ricordi che usando programmi cosiddetti di "SlideShow", tipo quello di Carolyne Sheppner o di Paul Biondo, appartenenti al "dominio pubblico" o comunque programmi americani di trattamento, si potrebbero avere piccoli problemi.

Questo perchè Butcher funziona in PAL, con 256 e 512 linee orizzontali contro le 200 e 400 americane. Perciò, ad esempio, una figura potrebbe apparire in modo interlacciato (sfarfallante) da uno dei programmi citati perchè lo standard americano prevede, al massimo, 200 linee; se un nostro disegno ne possiede 256, ovviamnete, passa a mostrarlo in modo interlacciato.

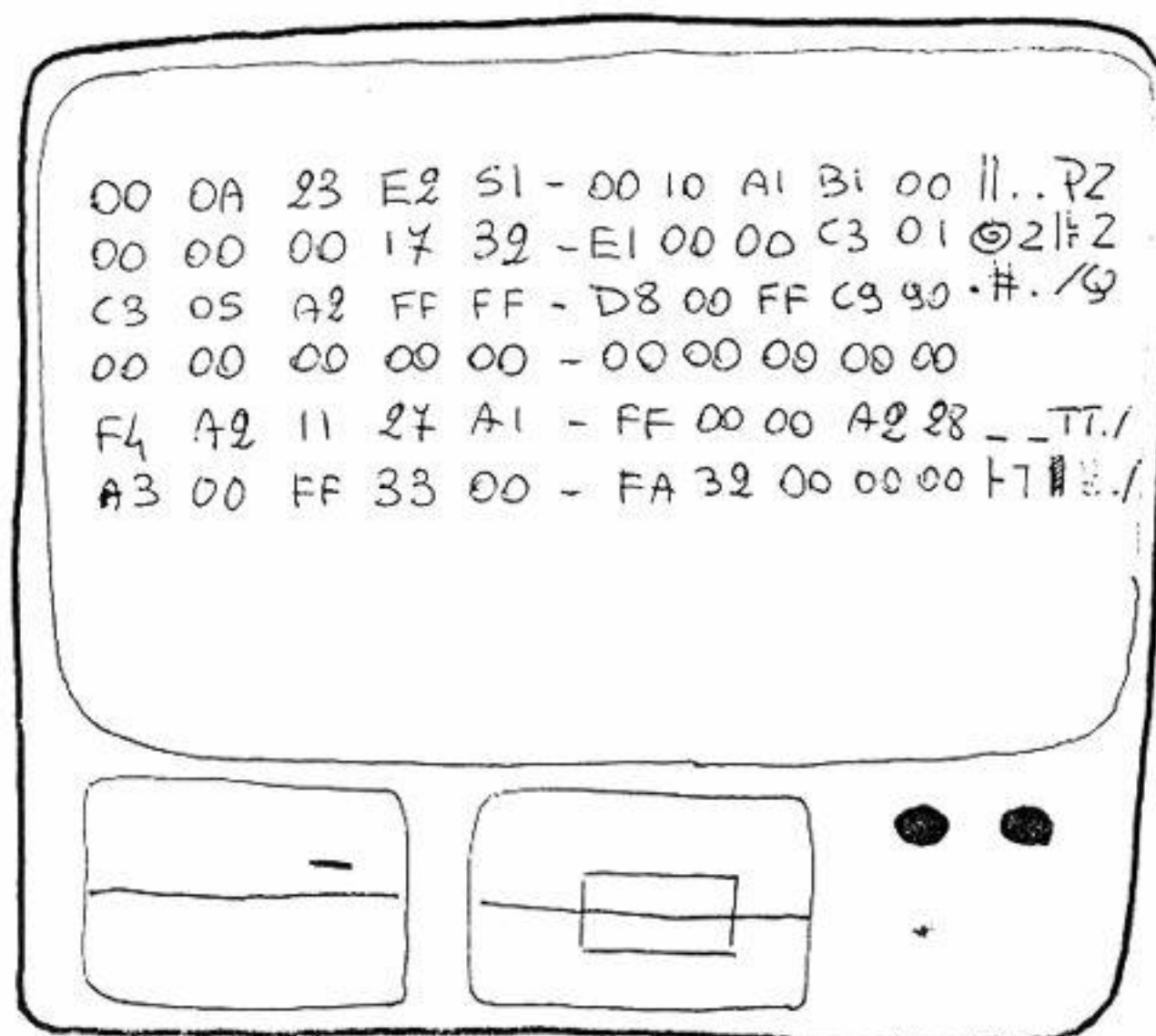
Il difetto può essere evitato correggendo, tramite l'opzione FORMAT del menu PROJECT, il numero di linee, oppure ricaricando il file in Deluxe Paint e risalvandolo col giusto formato.

Usando le varie finestre di Butcher, attivate da numerose opzioni sofisticate, appaiono spesso opzioni identiche, che spieghiamo per i più impazienti:
RESTORE solitamente annulla il lavoro fatto da quando si è attivata l'opzione, recuperando l'immagine originaria.
CANCEL simile alla precedente, cancella l'ultima operazione ma non necessariamente recupera l'originale.
KEEP rende effettive le modifiche fatte al disegno da quando si è iniziato a lavorare da un certo menu di finestra.
CLIP fissa una porzione rettangolare dello schermo, indicata col mouse (pigiare tasto sinistro, mantenere pigiato e muovere; poi, rilasciare) come l'unica interessata al lavoro. Tale porzione rimane attiva finchè non si clicka di nuovo sul rettangolino di CLIP.

Quasi tutte le opzioni possono essere interrotte usando la barra spaziatrice, oppure clickando sull'icona di una finestra di opzioni visualizzata in quel momento.

Per annullare completamente l'operazione è però spesso necessario ricorrere a UNDO.

ULTIME NOVITA'

Mentre scriviamo, abbiamo visionato gli ultimi prodotti grafici per Amiga. Attualmente un certo "Photon Paint" permette di disegnare come Deluxe Paint (salvo qualcosa), ma anche in HAM, mentre in formato standard ammette sino a 64 colori invece dei tipici 32. Altre novità rilevanti sono visibili con programmi tipo Pixmate e Express Paint; ma ne riparleremo.

I SEGRETI DEL C/16

Ecco una manciata di "segreti" del C/16 (e Plus/4), da utilizzare nei vostri programmi, siano essi scritti in Basic o in Assembly

di **Antonio Pastorelli**

REGISTRI DEL PROCESSORE

Le locazioni 2034, 2035, 2036 contengono, rispettivamente, l'accumulatore ed i registri X e Y del microprocessore.

Utilizzando queste locazioni è possibile assegnare parametri, ai registri del processore, prima di utilizzare routine in linguaggio macchina attivabili da Basic.

Ad esempio la routine "CHROUT" del sistema operativo (S.O.), che provvede ad inviare dati in uscita verso un canale, richiede che il dato da spedire sia contenuto nell'accumulatore.

Dal momento che il canale di uscita di default è il video, per stampare una "A" sullo schermo, basterà porre 65 (codice ascii di "A") nell'accumulatore, quindi chiamare la routine "CHROUT", come nel seguente esempio:

```
Poke 2034,65: Sys 65490
```

Per altre routine, che richiedono anche il preventivo caricamento dei registri X ed Y con determinati valori, ci si comporterà in modo del tutto simile a quello appena visto. Ricorriamo ad un esempio: la routine PLOT del S.O. svolge due funzioni, assegna le coordinate del cursore o le legge.

Questa routine richiede che la coordinata X sia posta nel registro Y, mentre la coordinata Y deve essere assegnata al registro X; se vi chiedete il perchè di questo strano scambio, sappiate che ce lo chiediamo anche noi!

In ogni caso, assegnate le coordinate ai registri Y ed X, basterà chiamare la routine del S.O. affinchè queste vengano accettate:

```
Poke 2036,10: Poke 2035,2: Sys
55355: Print "pippo"
```

I quattro comandi scrivono "PIPPO" sulla seconda linea, a partire dalla decima colonna.

LISTARE O OCCULTARE?

Ritornando sullo scottante tema delle protezioni, vi proponiamo un metodo (forse già noto) per evitare il List dei vostri programmi.

Le locazioni 774 e 775 rappresentano il vettore di List; ciò significa che, ogni volta che si impartisce il comando List, l'interprete Basic salta all'indirizzo contenuto nelle locazioni contigue 774 / 775.

In situazione di default, tali locazioni contengono i valori 110 e 139, puntando quindi a 35694 (= 110 + 139 * 256), a partire dalla quale locazione si trova la routine che lista il programma e traduce i Token in parole chiave del Basic.

Se, quindi, cambiamo l'indirizzo puntato dal vettore di List, facendolo puntare, ad esempio, alla routine di Power Up, avremo come risultato il Reset del computer se digitiamo il comando List.

Dal momento che la routine di Reset è allocata a partire da 62116,

per ottenere l'effetto Reset bisognerà impartire i seguenti comandi:

Poke 774,164: Poke 775,242

Infatti 164 + 242 * 256 = 62116. Volendo evitare il Reset, ma lasciar nascosto il listato Basic, battete: Poke 774,57; digitando, in seguito, List, si otterrà la visualizzazione dei soli numeri di linea.

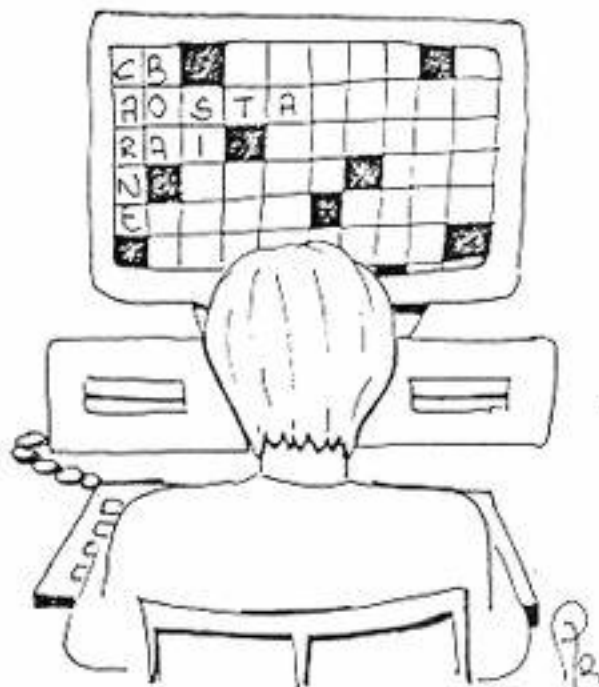
VETTORE MESSAGGI D'ERRORE

Le locazioni 768-769 contengono l'indirizzo della routine che stampa i messaggi d'errore.

La routine, che parte da 34438, stampa il messaggio d'errore il cui codice è contenuto nel registro X. Per farlo stampare bisogna caricare il registro X con il codice del messaggio e lanciare la relativa Sys...

Poke 2035, Codice errore: Sys 34438

Ecco, di seguito, la tabella dei codici degli errori:

1 Too many files
2 File open
3 File not open
4 File not found
5 Device not present
6 Not input file
7 Not output file
8 Missing file name
9 Illegal device number
10 Next without for
11 Syntax
12 Return without gosub
13 Out of data
14 Illegal quantity
15 Overflow
16 Out of memory
17 Undef'd statement
18 Bad subscript
19 Redim'd array
20 Division by zero
21 Illegal direct
22 Type mismatch
23 String too long
24 File data

25 formula too complex
26 Can't continue
27 Undef'd function
28 Verify
29 Load
30 Break
31 Can't resume
32 Loop not found
33 Loop without do
34 Direct mode only
35 No graphics area
36 Bad disk

Esempio:
Poke 2035,28: Sys 34438

...stampa il messaggio "Load Error".

Attenzione: la routine d'errore, dopo la stampa del messaggio, salta alla routine di READY, pertanto non è possibile proseguire con il programma, Basic o Assembler che sia. Può essere utile in alcune tecniche di protezione in cui si desidera far apparire un errore, in realtà inesistente.

L'ULTIMA OPERAZIONE

La locazione 147 è un flag che viene aggiornato dopo ogni operazione da/verso la periferica.

Se il suo contenuto è nullo, allora l'ultima operazione eseguita (o tentata), è stata un caricamento; in caso contrario si è trattato di una verifica.

SENZA BUFFER E' MEGLIO

Quando alcune operazioni interrompono un flusso di dati, per un determinato periodo di tempo, il computer "bufferizza" i dati stessi, per poi considerarli quando ha terminato l'operazione in corso.

E' questo il caso della gestione della tastiera; anche se il computer è impegnato in calcoli (od altre operazioni), premendo un tasto, il suo codice viene immagazzinato in un'area Ram temporanea, ed al termine dell'operazione viene ripreso per essere opportunamente trattato.

Questa tecnica, molto utile nella maggior parte dei casi, diventa fastidiosa in altri.

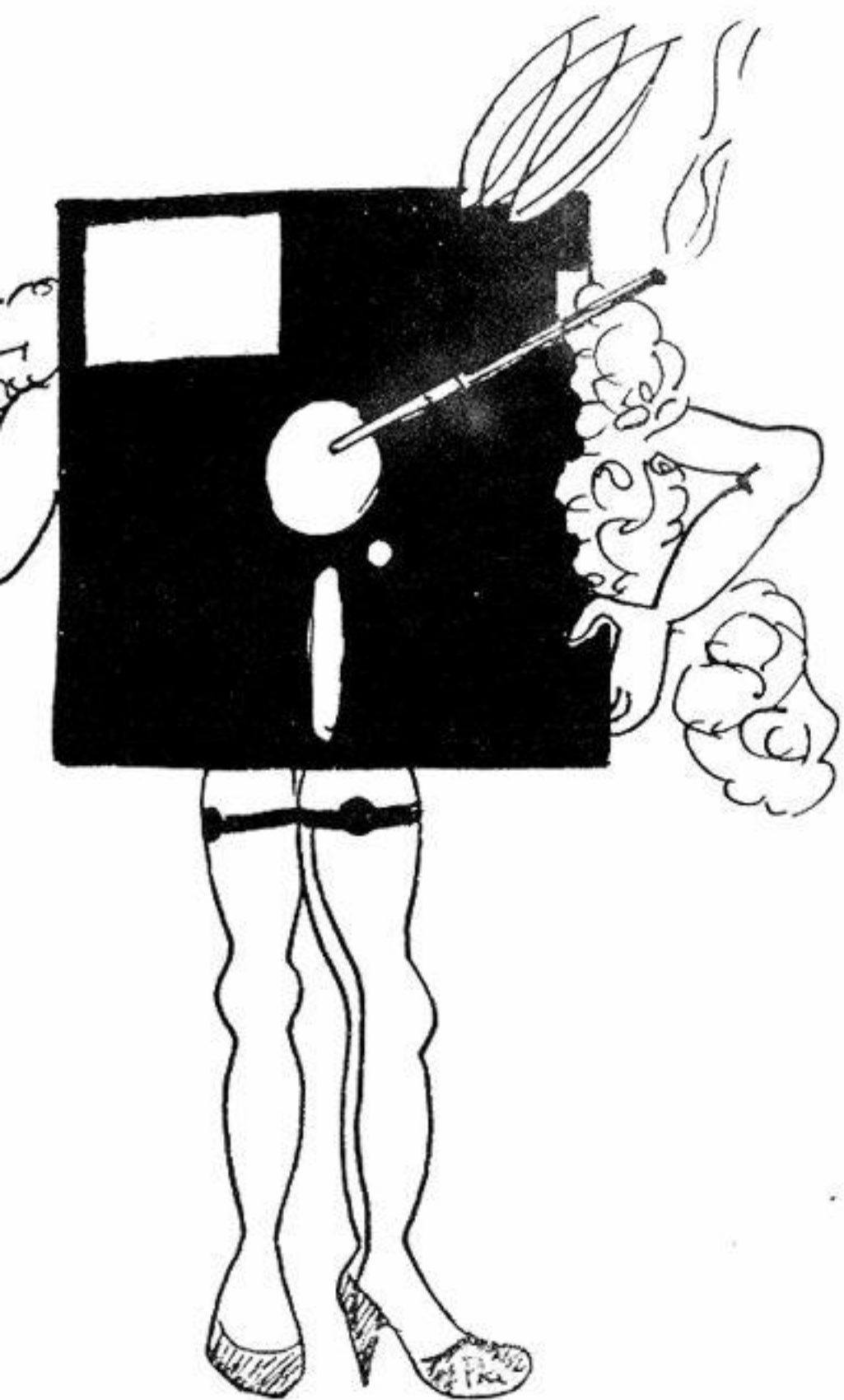
Se, ad esempio, abbiamo creato un programma il cui funzionamento è gestito da menu nidificati, per i quali le scelte vengono accettate da tastiera, può capitare che, premendo dei tasti durante le attese, questi vengano memorizzati ed in seguito riconosciuti come scelte relative ai menu stessi, operando quindi scelte non volute.

Per meglio comprendere il discorso, digitate le seguenti linee:

```
10 for j=1 to 10000
20 next j
30 print "premi un tasto per continuare!"
40 get a$: if a$="" then 40
50 print "ok."
```

Date il Run, seguito da Return, poi premete un tasto qualsiasi.

All'apparire del messaggio "Premi un tasto per continuare", verrà stampato "OK." ed il programma terminerà, anche se alla linea 40 si attende la pressione di un tasto per continuare. Ciò succede perchè il tasto premuto dopo il Run è stato immagazzinato nel Buffer e riutilizzato per la richiesta di pressione di un tasto. Fate ripartire il programma e, durante l'attesa del ciclo For... Next, premete molti tasti a caso. Vedrete meglio l'effetto che ne deriva.

Per evitare questo comportamento, basta azzerare la locazione 239 che contiene il numero di caratteri bufferizzati. Inserendo un'istruzione Poke 239,0, prima di ogni GET da tastiera, si evitano problemi di scelte non desiderate, dando una veste più professionale ai vostri programmi.

QUALE TASTO?

Un metodo veloce e sicuro, per prelevare dati dalla tastiera, è quello di leggere il valore della locazione 198.

Tale locazione contiene costantemente il codice del tasto premuto, altrimenti indica 64.

I codici di cui parliamo sono relativi ai tasti e non alla loro codifica in ASCII; leggendo la locazione 198 otterrete lo stesso valore premendo "B" oppure i tasti Shift e "B": il codice memorizzato è sempre lo stesso!

I TASTI NASCOSTI

Normalmente, quando si prelevano dati da tastiera, non è possibile intercettare la pressione dei tasti Shift, Commodore e Control, in quanto vengono utilizzati sempre unitamente ad altri tasti.

La locazione 1347 svolge la stessa funzione che la 198 svolge per gli altri tasti; rileva, infatti, la pressione dei tasti citati anche congiuntamente. I valori ottenuti leggendo tale locazione vanno interpretati in accordo con la seguente tabella:

0 Nessuno
1 Shift
2 Commodore
3 Shift e Commodore
4 Control
5 Shift e Control
6 Commodore e Control
7 Shift, Commodore e Control

Esempio d'uso:

```
10 Print peek (1347): goto 10
```

Lanciate il miniprogramma e premete i tasti in questione, prima uno per volta, poi congiuntamente, e notate i valori stampati sul video.

L'OROLOGIO INTERNO

L'ora, che viene azzerata all'accensione del calcolatore, è costantemente aggiornata nelle locazioni 163 - 164 - 165 ed è espressa in sessantesimi di secondo; la locazione 163 contiene il byte alto (X 65536), 164 contiene il byte medio (X 256) e 165 contiene il byte basso (X 1).

SIMULA CMD

L'uscita di default dei dati è il video, e può essere cambiata tramite il comando CMD.

La locazione 153 contiene il codice della periferica di output.

Caricate in memoria un piccolo programma Basic, poi digitate quanto segue:

```
Open 1,8,12, "Prova,s,w": Poke 153,8: List: Close 1
```

Otterrete il listato del programma su disco, anzichè su video, in formato ASCII; potrà quindi essere caricato con un Word processor, e trattato come un qualsiasi file di testo.

Per listare su stampante si sostituirà Poke 153,8 con Poke 153,4.

IN CHE STATO SEI?

La variabile di stato, ST, che indica lo stato del computer al termine di determinate operazioni, è memorizzata nella locazione 144 che può essere utilizzata in alternativa alla variabile ST.

MA CHE MODO E' QUESTO?

Per conoscere il modo grafico attualmente selezionato, è possibile leggere la locazione 131, tramite la quale è anche possibile cambiarlo.

Tabella modi grafici:

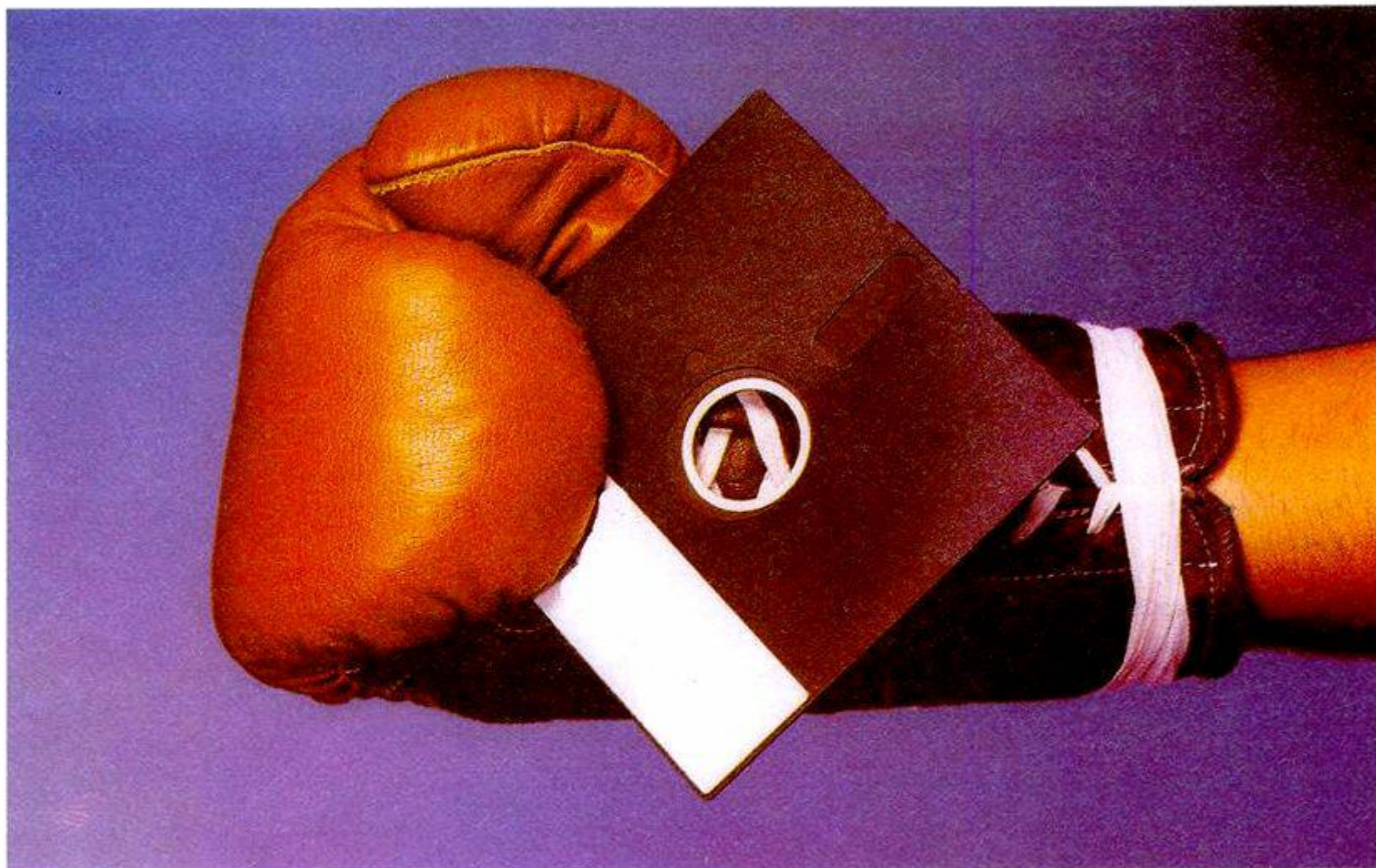
0 Modo testo
32 Alta risoluzione
96 Alta risoluzione con finestra testo
160 Multicolor
224 Multicolor con finestra testo

SCROLLARSI LO SCROLL

La combinazione di tasti Control + S ferma l'eventuale scrolling del video. Se, ad esempio, facciamo girare il programma seguente...

```
10 a=a+1: print a: goto 10
```

...premendo Control contemporaneamente al tasto "S", otterremo non solo l'arresto dello scroll, ma anche il blocco del programma stesso, dal momento che la sua unica funzione è quella di stampare su video.

Per disabilitare la possibilità di arrestare lo scrolling, bisogna porre ad 1 la locazione 2039. Per ristabilire le condizioni normali, invece, si azzererà la stessa locazione.

VIRUS-MANIA

Finalmente alla portata di tutti un argomento tra i più interessanti dell'informatica, i virus. Vedremo come realizzarli, renderli innocui e, perchè no, addomesticarli...

di **Michele Maggi**

Il fenomeno Virus non è nuovo anzi, è vecchio quasi quanto l'informatica.

Il fatto che se ne parli molto in questo periodo è dovuto in gran parte alla presenza di virus su moltissimi programmi per Amiga e Ibm compatibili.

In realtà il virus ha radici ben più lontane e all'inizio era, per così dire, rischioso solo per i grossi sistemi professionali; a questo proposito sarebbe utile rileggere l'articolo "Verrà la morte e sarà un virus" pubblicato su C.C.C. n. 49.

Ma che cos'è e come nasce un virus?

Il virus è l'esasperazione dei sistemi di protezione, quasi una vendetta postuma che si rivolge contro chi utilizza software copiato.

La necessità di creare un virus nasce dell'evidente impotenza di proteggere adeguatamente il software prodotto; inutile dire che esistono in commercio copiatori sempre più sofisticati che sono in grado di annichilire anche la più potente protezione. Perfino il "laser-hole", la fantastica protezione al laser, è ormai da considerare superata.

Quando non si riesce a trovare un mezzo di protezione efficace contro la copiatura, viene scatenata l'infe-

SCHEDA TECNICA

Software didattico per l'applicazione di protezioni insolite

Hardware richiesto: C/64, C/128 (modo 64). Non adattabile ad altri computer Commodore

Consigliato ai principianti

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

LA PROTEZIONE LASER-HOLE

Nell'esasperata ricerca di metodi di protezione efficaci si è giunti, qualche anno fa, al famoso "laser-hole" un sistema rivoluzionario che avrebbe dovuto rappresentare la soluzione finale ai problemi di copiatura illegale.

Il suo funzionamento si basa su un forellino praticato, grazie all'impiego di un raggio laser, sulla superficie del dischetto e da un'opportuna routine, presente nel programma, che ne controlla, o meno, l'esistenza.

La peculiarità del sistema è che mentre un qualsiasi errore può essere riprodotto perchè risultante da un "disordine" particolare delle zone magnetiche del disco, il foro laser non è riproducibile se non tramite complesse apparecchiature.

Questo sistema di protezione (riservato peraltro a dischetti per sistemi Ms-Dos compatibili) si è rivelato efficace per lungo tempo, fino all'introduzione di nuovi copiatori (come il Copywrite 5) che, anche se molto lentamente, sono in grado di simulare il laser-hole e ciò ad ennesima dimostrazione che ogni metodo di protezione, anche se all'avanguardia e rivoluzionario, non sarà mai totalmente sicuro.

zione virale, operazione forse drastica, ma necessaria.

Il virus, una volta attivato, infetterà una zona di memoria del computer (naturalmente solo fino al suo spegnimento; pensate, tuttavia, a quei computer dotati di memoria costante, permanentemente alimentata da batterie tampone...) e quindi si trasmetterà a tutti i dischi che verranno inseriti o, peggio, all'hard disk; se sono in atto comunicazioni via modem, anche il computer ricevente rischierà di essere infettato con il conseguente dilagare dell'infezione.

Da un punto di vista etico il virus si può considerare come un eccesso di legittima difesa, ma purtroppo questo metodo di difesa non conosce mezze misure; una volta attivo si può trasmettere via modem anche ad un utente che magari non ha mai utilizzato un programma copiato. Ma si sa, la storia è piena di vittime innocenti...

A discolpa del virus bisogna dire che esso raramente agisce al primo contatto; di solito si sveglia dopo una decina di utilizzazioni del programma duplicato illegalmente; ciò significa che l'utente avrebbe tutto il tempo di pentirsi e di distruggere il software copiato.

Chi sono gli "untori" che diffondono l'infezione? Ne esistono due categorie: coloro che ne fanno uso per tutelarsi e coloro che lo usano per divertimento, al solo scopo di produrre danni, almeno a dar credito alle notizie riportate di recente sulla diffusione di virus in molti centri di ricerca ed universitari.

LA NASCITA DI UN VIRUS

Dal punto di vista tecnico un virus altro non è che un programma, magari di poche centinaia di byte, che gira indipendentemente da ciò che il computer sembra elaborare.

La tecnica più indicata sarà quindi quella di sistemare la nostra routine - virus all'interno di un interrupt (o in un task, per i computer multitasking).

DATA SOTTO CONTROLLO

Come di consueto i listati con numerose linee Data rappresentano un problema, soprattutto per i principianti, in quanto ogni numero indica un codice assembler da trascrivere con esattezza.

Il controllo di checksum (linea 4) ha lo scopo di controllare la somma dei numeri Data, ma non sempre è efficace in quanto allo stesso totale possono concorrere diversi addendi; ad esempio 4 + 6 fa 10, ma anche 2 + 8 oppure 5 + 5.

Ciò significa che se, copiando i data, modificate, per errore, un numero che "riequilibrate" in un numero successivo, il risultato totale sarà il medesimo ma i valori saranno errati.

Questo discorso vale solo per i numeri tra 0 e 255, oltre i quali si verificherebbe un Illegal quantity che causerebbe l'interruzione del programma prima della routine di controllo.

Per ovviare a ciò la linea 3 tiene conto del numero di linea Data elaborata, in modo da indicarlo se si verificasse un errore; i byte 63 e 64 contengono, infatti, in numero di linea Data corrente.

Se il computer dovesse quindi segnalare un Illegal quantity error, basterà controllare la linea Data visualizzata sullo schermo..

Nel programma definitivo, ovviamente, vanno eliminate le linee di controllo (3, 5, 6) non solo perchè superflue ma, soprattutto, per eliminare ogni sospetto di azione... insolita nell'ignaro utilizzatore.

E' ovvio che, al posto del semplice programmino (100 - 250) per determinare la media di dieci numeri, il lettore dovrà provvedere ad inserire il programma da "proteggere". Altrettanto ovvia è la possibilità di decidere la parola d'ordine, alterando con la massima semplicità quella riportata nella riga 9 del programma proposto.

Agendo in interrupt il nostro virus agirà indisturbato, e di nascosto, svolgendo con discrezione il suo lavoro.

Vediamo ora come è possibile creare un mini-virus per il nostro C/64 ma, niente paura: sarà del tutto innocuo, riportato a puro scopo dimostrativo per meglio capire come può funzionare questo lato nascosto dell'informatica.

INFETTIAMO IL C/64

Supponiamo che abbiate creato un programma interessante, del quale andate molto fieri, che deve "girare" solo in vostra presenza.

In questo caso la protezione tipica è la password dal momento che solo l'utente autorizzato può essere in grado, conoscendo il codice d'accesso, di utilizzare il programma.

Il successivo ricorso ad un compilatore (tipo Austrospeed), nel caso il software sia scritto in basic, è d'obbligo, anzi per le routine di "infezione" è sempre meglio ricorrere al basic compilato che spesso risulta più difficile da decifrare anche per gli smanettoni.

La routine virus sarà in L.M. per ovvi motivi di velocità e compattezza.

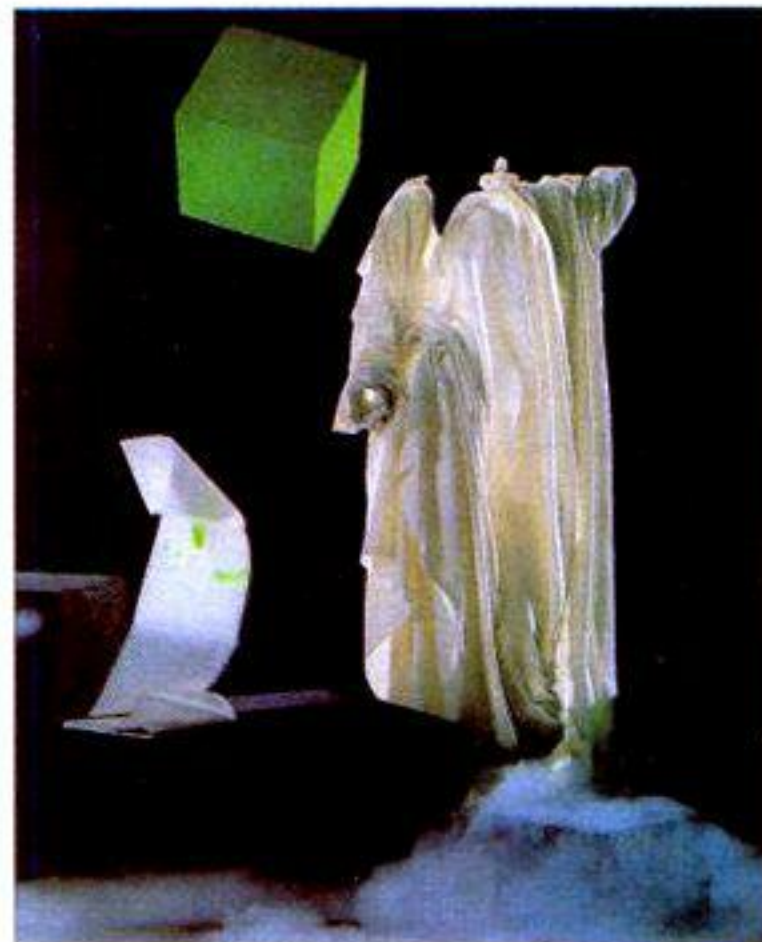
A questo punto, una volta creato il virus, vediamo come inocularlo nel computer. La prima cosa da fare sarà allocarlo in una zona di memoria "libera" da interferenze con il programma principale; si può ricorrere alla consueta sequenza di Read... Data che, una volta compilata, sarà molto veloce e darà il tempo per una presentazione del programma in modo da distrarre l'utente.

Una volta allocato il virus bisognerà valutare se attivarlo oppure no; ciò dipenderà dalla parola chiave digitata: se viene inserita quella esatta, il programma funzionerà normalmente; se ne viene inserita una diversa il programma funzionerà ugualmente, ma con una differenza: il virus verrà attivato!

Il fatto di far partire il programma qualunque sia la parola d'ordine digitata, serve a non insospettire l'utente pirata che, altrimenti, potrebbe prendere opportuni provvedimenti.

A questo punto il nostro virus potrà svolgere il compito affidatogli, tra cui: cancellare la memoria, indurre il computer a fornire risultati errati, formattare i dischi a sorpresa(!) ed altre piacevolezze.

Nell'esempio riportato in queste pagine il virus si limiterà a far scorrere lungo lo schermo uno sprite che cambierà colore ad ogni contatto con i caratteri, quasi si trattasse di un fastidioso batterio che, senza danneggiare nulla, si limiterà a manifestare la sua presenza; ma ricordate, se al posto di uno sprite che se ne va in giro tranquillamente avessimo utilizzato una routine di formattazione, le conseguenze sarebbero state ben più evidenti...

Data la tecnica usata, possiamo dire che ogni programma in interrupt è potenzialmente un virus: dipende solo da ciò che si decide di fargli fare.

VIRUS: ISTRUZIONI PER L'USO

In queste pagine, a parte il disassemblato, sono presenti due listati, fusi tra loro in uno solo: il primo è il loader del virus (righe da 1 a 10 e da 63979 a 63998) mentre il secondo (100 - 250) è un programma per il calcolo della media di dieci numeri; a dispetto della sua banalità, ha lo scopo di far capire in che modo il lettore può inserire il virus in un proprio programma.

Se alla richiesta della password si risponde con "MK64" (riga 9) il programma continuerà normalmente (Goto 100); in caso contrario, il virus verrà attivato (riga 10) ed inizierà il suo lavoro.

Il metodo della password è solo uno dei sistemi per attivare, o meno, un virus; prossimamente vedremo come far scattare un virus solo dopo un certo numero di usi illegali di un programma e, soprattutto, come renderlo resistente ai due "antibiotici" possibili: ci riferiamo non solo al Run / Stop Restore ma anche al Reset.

UNO SGUARDO AL DISASSEMBLATO

Il programma assembler è allocato a partire da 49152 ($C000), zona lontana dal basic e raramente usata in programmi che non facciano ricorso al linguaggio macchina.

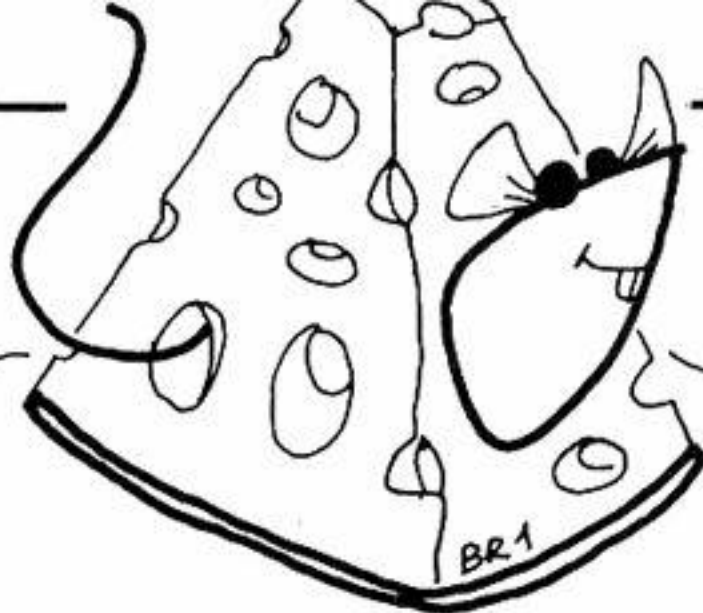
Le linee 7 - 12 dirottano i vettori di interrupt e il blocco 13 - 19 crea uno sprite quadrato traducendo, in L.M, la riga basic:

For i=0 to 63: poke 832 + i, 255: next

Vengono poi settati il puntatore agli sprite (2040), quello di accensione e le coordinate di partenza.

Il resto, a parte la routine di raster che ha il solo scopo di evitare sfarfallamenti, si occupa del movimento periodico dello sprite di cinque pixel per l'asse X e di tre per Y ogni volta che esce dallo schermo.

Nulla di particolare da dire circa la routine di collisione (linee 54 - 56) che agisce sul byte 53279 ($D01F).

```
1 REM CARICATORE VIRUS BY MICHELE MAGGI
2 PRINT CHR$(147): FOR I=0 TO 125: READ A: CK=CK+A
3 PRINT CHR$(19); PEEK(64)*256+PEEK(63)
4 POKE 49152+I,A: NEXT:IF CK=16214 THEN 6
5 PRINT "ERRORE NEI DATA...": END
6 PRINT CHR$(147) "VIRUS OK..."
7 POKE 808,234: REM DISABILITA RUN/STOP
8 INPUT "PASSWORD";PS$: REM RICHIESTA CODICE
9 IF PS$="MK64" THEN 100: REM VIRUS INATTIVO
10 SYS 49152:REM ATTIVAZIONE VIRUS
11 :
12 :
13 :
100 REM ********************************
110 REM * ESEMPIO DI ATTIVAZIONE VIRUS *
120 REM *      IN UN LISTATO BASIC.    *
130 REM *    E' POSSIBILE INSERIRE UN  *
140 REM *   PROPRIO LISTATO DA RIGA 11 *
150 REM * A 63978 LASCIANDO INALTERATE *
160 REM *   LE PRIME 12 RIGHE E QUELLE *
170 REM *    NUMERATE DA 63979 IN POI  *
180 REM ********************************
190 PRINT CHR$(5)
200 PRINT CHR$(147)"--CALCOLO DELLA MEDIA --"
210 T=0:FOR I=1 TO 10: INPUT"NUMERO";N
220 T=T+N:NEXT:PRINT"LA MEDIA VALE"T/10
230 PRINT: PRINT "PREMI UN TASTO"
240 GET A$: IF A$="" THEN 240
250 GOTO 200
260 :
270 :
63979 REM DATA PER VIRUS
63980 DATA 120,169,046,141,020,003,169
63981 DATA 192,141,021,003,088,169,255
63982 DATA 162,000,157,064,003,232,224
63983 DATA 063,208,248,169,013,141,248
63984 DATA 007,173,021,208,009,001,141
63985 DATA 021,208,169,040,141,000,208
63986 DATA 141,001,208,096,173,018,208
63987 DATA 201,255,208,249,173,000,208
63988 DATA 201,088,144,025,173,016,208
63989 DATA 240,020,173,016,208,073,001
63990 DATA 141,016,208,141,000,208,238
63991 DATA 001,208,238,001,208,238,001
63992 DATA 208,173,000,208,201,255,208
63993 DATA 008,173,016,208,073,001,141
63994 DATA 016,208,173,031,208,240,003
63995 DATA 238,039,208,238,000,208,238
63996 DATA 000,208,238,000,208,238,000
63997 DATA 208,238,000,208,076,049,234
63998 END
```

```
 1 *-------------------------------*
 2 * ESEMPIO DI INTRODUZIONE VIRUS *
 3 * CON UNA BREVE ROUTINE POSTA   *
 4 * NELL'INTERRUPT
 5 *-------------------------------*
 6
 7           SEI                ;DIROTTA
 8           LDA   #<ENTRY      ;I VETTORI
 9           STA   $0314        ;DI INTERRUPT
10           LDA   #>ENTRY      ;ALLA NUOVA
11           STA   $0315        ;ROUTINE
12           CLI                ;
13           LDA   #255         ;DEFINISCE
14           LDX   #0           ;IL DISEGNO
15 LOOP                         ;DELLO SPRITE
16           STA   $0340,X      ;E LO METTE
17           INX                ;NEL BUFFER
18           CPX   #63          ;DI CASSETTA
19           BNE   LOOP         ;
20           LDA   #13          ;SETTA IL
21           STA   2040         ;PUNTATORE
22           LDA   $D015        ;E ACCENDE
23           ORA   #$01         ;LO SPRITE
24           STA   $D015        ;UNO
25           LDA   #$28         ;SETTA LA
26           STA   $D000        ;POSIZIONE
27           STA   $D001        ;DI PARTENZA
28           RTS                ;ED ESCE
29 ENTRY
30 RASTER
31           LDA   $D012        ;CONTROLLA
32           CMP   #255         ;CHE IL RASTER
33           BNE   RASTER       ;SIA FUORI CAMPO
34           LDA   $D000        ;CONTROLLA
35           CMP   #88          ;LA X-POS
36           BCC   OUT          ;DELLO SPRITE
37           LDA   $D010        ;E SE E' OLTRE
38           BEQ   OUT          ;LO SCHERMO
39           LDA   $D010        ;AZZERA IL MSB
40           EOR   #1           ;E RIPORTA A
41           STA   $D010        ;A ZERO LA
42           STA   $D000        ;X-POS
43           INC   $D001        ;MUOVE IN GIU'
44           INC   $D001        ;LO SPRITE
45           INC   $D001        ;
46 OUT                          ;
47           LDA   $D000        ;SE LO SPRITE
48           CMP   #255         ;VA OLTRE
49           BNE   SKIP         ;255 IN X
50           LDA   $D010        ;VIENE SETTATO
51           EOR   #1           ;IL MSB
52           STA   $D010        ;
53 SKIP                         ;
54           LDA   $D01F        ;SE C'E'
55           BEQ   MOVE         ;COLLISIONE
56           INC   $D027        ;COLORA LO SPRITE
```

TI ACCENDERAI QUANDO LO DIRO' IO

Un paio di notizie, relative alla gestione del registratore, utili anche per secondi fini

Gli home computer della famiglia Commodore, tra cui il C/64, possiedono un connettore tramite il quale collegare il registratore dedicato.

Come tutti sapranno non è necessario inserire batterie o cavi di alimentazione, dal momento che il registratore preleva la tensione direttamente dal computer.

Questo, da parte sua, decide quando il motorino deve girare oppure restare spento, come si può notare impartendo un comando di tipo Load o Save (il motorino si accende), oppure premendo il tasto Run/Stop (si spegne).

Questa caratteristica può essere sfruttata a nostro vantaggio per applicazioni pratiche che vedremo in seguito; per ora cominciamo a capire come funziona il tutto.

Nel C/64 esistono due locazioni di memoria che servono per controllare il funzionamento del registratore: la locazione 0 e la locazione 1.

La n.0 è un registro di direzione dati, che decide quali informazioni devono entrare, oppure uscire, dal 6510 (circuito integrato dedicato a tali operazioni). I singoli bit hanno il significato indicato in tabella 1.

Se uno dei bit viene messo a 0 (resettato) significa che deve essere la memoria a fornire informazioni al microprocessore; mentre se è messo a 1 è il micro a fornire informazioni in uscita.

La locazione 1 è invece usata per controllare le varie partizioni di memoria RAM/ROM possibili sul C/64 ed anche per accendere e spegnere il motore del registratore.

I bit che la rappresentano hanno il significato riportato in tabella 2.

Come si può intuire, al programmatore interessano i bit 4 e 5, che

SCHEDA TECNICA

Articolo didattico per applicazioni h/w

Hardware richiesto: C/64, connettore per registratore e relè

Consigliato agli appassionati di applicazioni elettroniche

Si consiglia la massima prudenza nell'effettuare collegamenti elettrici

Bit	Direzione	Descrizione
0	Uscita	Controllo RAM/ROM $A000
1	Uscita	Controllo RAM/ROM $E000
2	Uscita	Controllo I/O ROM $D000
3	Uscita	Linea scrittura cassetta
4	Ingresso	Sensore int. registrat.
5	Uscita	Contr. motore registrat.

Tabella 1

controllano direttamente il motore del registratore.

Per "attivare" questi bit è necessario porre a 1 i corrispondenti bit della locazione 0 (cioè i bit 4 e 5), dato che, come detto prima, questa operazione farà in modo che sia il microprocessore a fornire dati all'esterno. Questo può essere fatto direttamente anche da Basic con un banale...:

```
POKE 0, 55
```

Se effettuate i calcoli opportuni, vedrete che il valore decimale 55 attiva proprio i bit interessati. Da questo momento sia il sensore che il motore del registratore sono pronti a ricevere comandi. Per far girare il motore dovremo porre a 1 anche i bit 4 e 5 della locazione 1, tramite...:

```
POKE 1, 55
```

Per "spegnerlo" si dovranno riportare a 0 i suddetti bit con il comando:

```
POKE 1, 7
```

Vediamo di capire a che cosa servono i due bit di cui stiamo parlando: il bit 4 della locazione 1 (sensore cassetta) è quello che controlla se è premuto un tasto sul registratore; impartendo, ad esempio, il comando Load, il computer stampa il messaggio "Press play on tape", ma lo schermo si spegne, e la lettura inizia, solo quando si preme il tasto PLAY sul registratore.

Il che significa che il computer viene, in qualche modo, informato della pressione di alcuni tasti sull'unità a cassette, e questo viene fatto tramite il bit considerato.

In pratica, ponendolo a 1, noi facciamo credere al 6510 che il tasto Play sia abbassato, e che quindi il motore può girare.

Il bit 5 della locazione 1 è quello che controlla direttamente la tensione da inviare al motore del registratore. Se è messo a 1, il C/64 metterà su un opportuno piedino della porta di uscita del registratore i 5 volt necessari ad attivare il motore, mentre se è messo a 0, sarà tolta la tensione ed il motore si fermerà.

Riassumendo: per avviare e spegnere il motore bisogna impartire anzitutto POKE0,55; poi, per accenderlo: POKE1,55; quando si vuole spegnere POKE1,7

Sapere tutto questo non è molto utile se non si trova un'applicazione pratica; naturalmente siamo qui per questo!

Dal dire al fare

Visto che, da quanto abbiamo detto, tramite semplici comandi è possibile avere, o meno, una tensione di 5 volt sulla porta del registratore, possiamo sfruttare questa tensione per comandare qualcosa di diverso dal motorino del registratore, magari il televisore o la lavatrice, o l'impianto HI-FI: ma in che modo?

Non certamente connettendoli direttamente al C/64, anche perchè quel tipo di apparecchi generalmente deve venire alimentato da una tensione di 220 volt, non propriamente compatibili con i delicati circuiti del nostro computer.

Come "interfaccia" tra i miseri 5 volt ed i possenti 220 ENEL, si può inserire un Relè.

Per chi non lo sapesse, un relè non è altro che un interruttore comandato elettricamente, cioè al suo interno è presente un'elettrocalamita. Se le viene fornita tensione l'interruttore sarà chiuso e permetterà, di conseguenza, il passaggio della corrente; mentre se viene tolta tensione all'elettrocalamita, quest'ultima rilascerà l'interruttore, riaprendolo, ed impedendo il passaggio della corrente (elementare, Watson).

Per far capire meglio ricorriamo ad un esempio concreto: supponiamo che siate in macchina e che, per un qualsiasi motivo, dobbiate frenare; pigiando il pedale del freno, la macchina rallenta, mentre rilasciandolo, quest'ultima cessa di rallentare.

In questa analogia il vostro piede

Bit	Descrizione
0	Segnale low ram
1	Segnale high ram
2	Segnale charen
3	Uscita dati per cassetta
4	Sensore cassetta (1=int. chiuso)
5	Motore cassetta (1=on, 0=off)
6,7	Non usati

Tabella 2

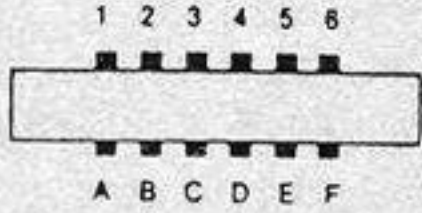
fa la parte dell'elettrocalamita che, tramite una tensione (la vostra "modesta" forza... pedestre) "chiude" l'interruttore, ossia consente l'azione frenante, e questo interruttore rimane chiuso fintantochè viene a mancare la tensione presente nell'elettrocalamita (viene rilasciato il pedale).

Il vantaggio del sistema è che il vostro piede è completamente isolato dall'impianto frenante; traducendo in termini elettronici, la tensione presente nell'elettrocalamita non influenza minimamente quella presente nell'interruttore, e viceversa.

Dato che, generalmente, l'elettrocalamita di un relè funziona a tensioni molto basse, ciò consente di comandare l'apertura e la chiusura del contatto tramite i 5 volt del C/64, senza preoccuparci della tensione che sarà presente ai capi dell'interruttore, che potrà quindi essere di 6, 12, 100, 220, 1000 volt ed oltre (a seconda del tipo di relè).

Ricapitolando, non dobbiamo fare altro che collegare i capi della bobina del relè agli opportuni contatti del connettore del registratore (dal lato del computer) e poi collegare l'apparecchio da accendere o spegnere al comando all'interruttore del relè.

I contatti della porta del registratore che interessano sono il n.1 ed il n.3 (guardando il computer dal retro, il primo ed il terzo partendo da sinistra). Il contatto 1 è la massa, mentre il 3 è quello che porta la tensione per il motore del registratore.

Impartendo POKE1,55 (ricordate, però, di fare prima di tutto POKE0,55), tra il piedino 1 ed il 3 sarà presente una tensione di 5 volt che, passando attraverso l'elettrocalamita, farà scattare l'interruttore del relè.

Siccome il nostro apparecchio è collegato all'interruttore, questo si accenderà, spegnendosi solo quando sarà inviato un POKE1,7 che, togliendo tensione all'elettrocalamita, farà scattare in senso inverso l'interruttore, impedendo il passaggio della corrente che alimentava il nostro utilizzatore.

L'unico svantaggio di questo sistema è che è possibile comandare una sola presa di corrente, e quindi un solo apparecchio (oppure più di uno, ma che si accende e si spegne allo stesso modo di quello già collegato).

Per comandare più apparecchi contemporaneamente in modo indipendente l'uno dall'altro, si può sfruttare la user port, che possiede ben 8 contatti utilizzabili e quindi 8 relè che comandano altrettanti apparecchi.

In questo caso, comunque, la storia diventa un po' più complicata, dato che la debole corrente in uscita dalla user port non basterebbe a far scattare un relè, rendendo necessario l'uso di un transistor di potenza o amenità del genere.

Pin	Type
A-1	GND
B-2	+5V
C-3	CASSETTE MOTOR
D-4	CASSETTE READ
E-5	CASSETTE WRITE
F-6	CASSETTE SENSE

1 2 3 4 5 6

A B C D E F

A CACCIA DI AVVENTURE

Scrivere un programma di adventure, in Basic, è piuttosto semplice. Anche i principianti possono giungere a strepitosi risultati con il minimo impegno

di **Alessandro de Simone**

ANCORA PIU' AVANTI

I programmi pubblicati, nonostante rappresentino un gioco "completo", risultano piuttosto limitati per ciò che riguarda strategia, difficoltà e trama.

Ciò che importa sottolineare, tuttavia, è la notevole sofisticazione che, volendo, si può raggiungere.

E' possibile, tra l'altro, assegnare anche altri valori (come, ad esempio, "3") per individuare caselle che contengono premi o, al contrario, trappole mortali; ed aggiungere righe basic che agiscano di conseguenza.

Invitiamo pertanto i lettori più attenti, pur se principianti, ad esaminare con attenzione i listati pubblicati e ad aggiungere le variazioni desiderate.

Si tenga presente, in ogni caso, che volendo introdurre numerosi codici (oltre ai valori 0, 1 e 2 dei programmi pubblicati) si rischia di cancellare involontariamente alcuni codici già memorizzati in precedenza.

Supponiamo, in altre parole, che alla coordinata X=4, Y=7 sia associato il numero 2 (obiettivo). Una successiva operazione random, con cui determinare le coordinate dell'elemento in cui depositare il codice 3 (ad es. un premio da prelevare) potrebbe assegnare le stesse coordinate 4 e 7. In questo caso la matrice risulterebbe priva dell'obiettivo con la conseguenza di vagare in lungo e in largo senza giungere alla soluzione del gioco.

La necessità di evitare simili inconvenienti (ed altri casi particolari di notevole importanza) costringerà piacevolmente i nostri lettori a sudare parecchio prima di pervenire a risultati degni di nota!

Ognuno di noi ha sicuramente visto, almeno una volta, un ADVENTURE.

A tale categoria appartengono i giochi di pazienza in cui il protagonista, avventurandosi in un labirinto dai contorni più o meno precisi, ha lo scopo di raggiungere un obiettivo.

I vari giochi in circolazione mettono, a disposizione dell'utente, armi, denaro, strumenti vari; inoltre, durante lo svolgimento, si possono fare brutti incontri, sbucare in luoghi in cui sono presenti oggetti da prendere (o da lasciare). Spesso, nelle versioni più recenti, il gioco si interrompe per cambiar ge-

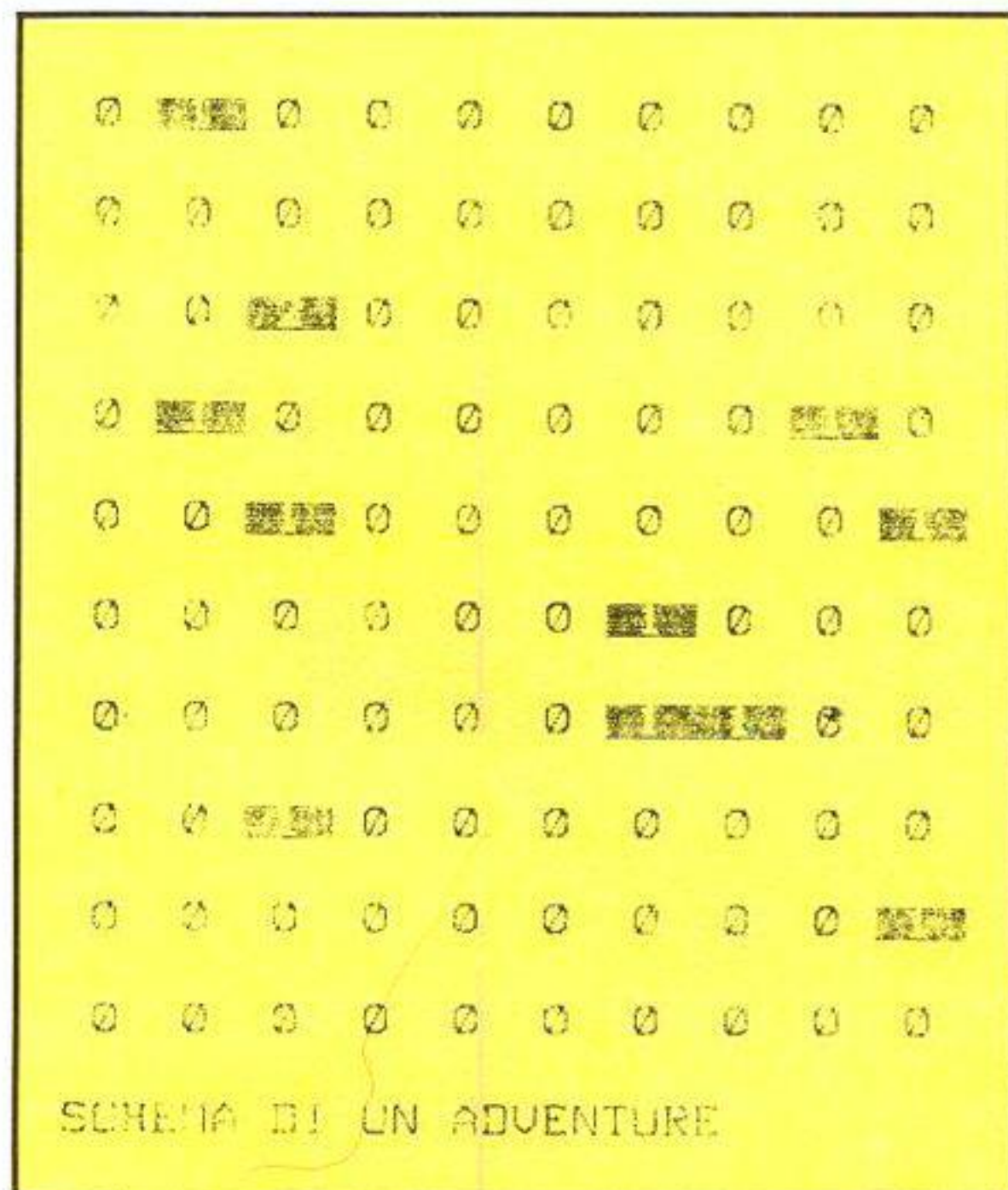
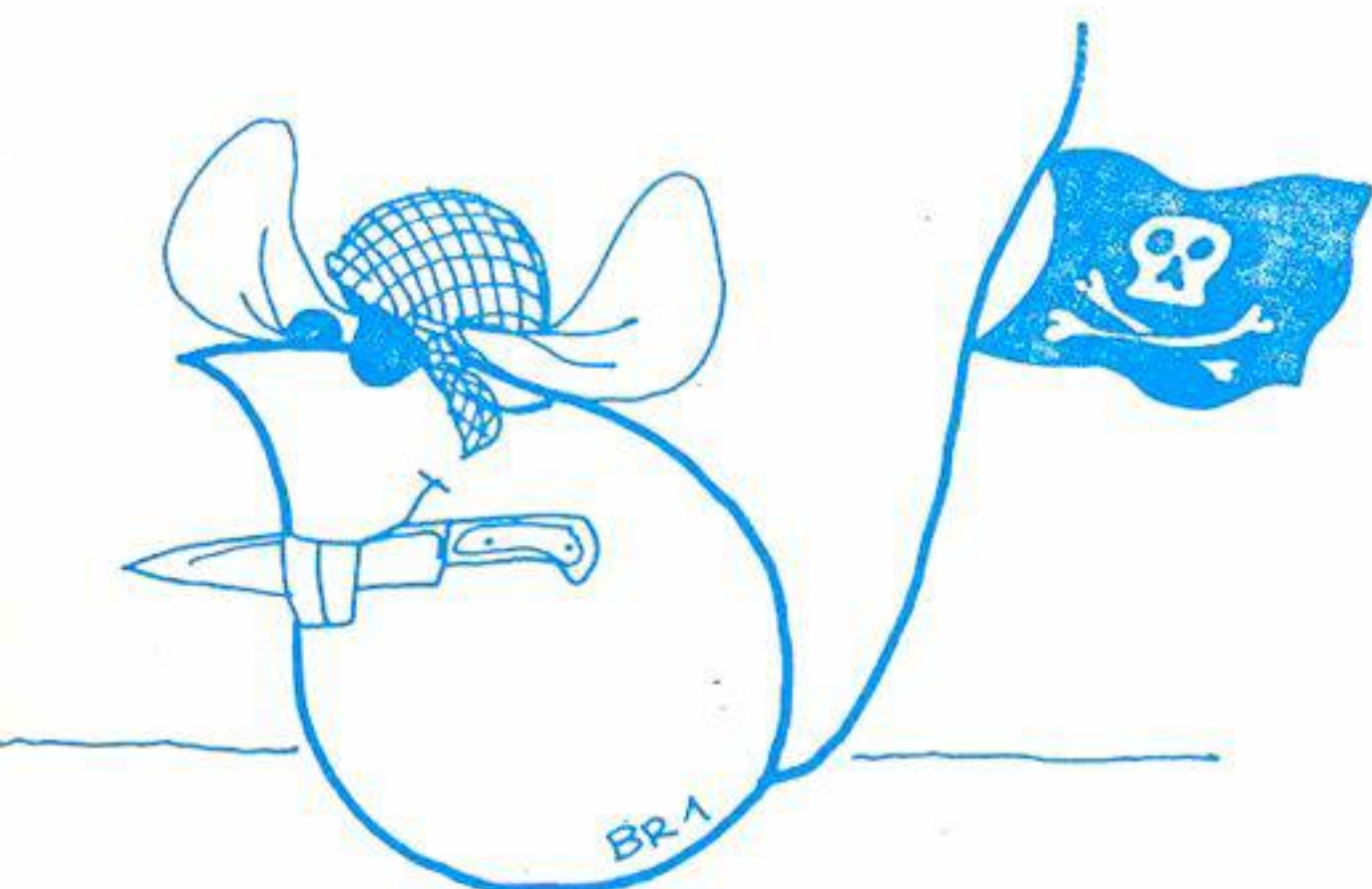
Poche righe Basic sono più che sufficienti per realizzare un semplice gioco di avventura

LE AVVENTURE DI

PRIMO GIOVEDINI

by M. Mietta

Il sogno di Primo

```
0 ██ 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 ██ 0 0 0 0 0 0 0
0 ██ 0 0 0 0 0 0 ██ 0
0 0 ██ 0 0 0 0 0 0 ██
0 0 0 0 0 0 ██ 0 0 0
0 0 0 0 0 0 ██ ██ 0 0
0 0 ██ 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 ██
0 0 0 0 0 0 0 0 0 0
SCHEMA DI UN ADVENTURE
```

Il "percorso" è il problema più importante, ma anche il più semplice, in un adventure.

nere: si visualizzano animazioni, spesso in alta risoluzione, in cui il giocatore, usando il joy, combatte contro nemici terribili oppure si destreggia in varie abilità.

In ogni caso, alla base di un qualsiasi adventure, è presente un "PERCORSO", di solito bidimensionale, da percorrere in lungo e in largo grazie ad i comandi E (Est), O (Ovest), N (Nord) e S (Sud).

Si giunge alla fine del gioco quando si individua il fatidico "OBIETTIVO" evitando i numerosi "OSTACOLI" di fronte ai quali è necessario prendere opportuni provvedimenti.

IL PERCORSO

Se (almeno all'inizio!), si rinuncia alle animazioni grafiche, alle musichette di sottofondo e ad altre sofisticazioni del genere, creare un percorso è un'operazione decisamente semplice.

La figura di queste pagine rappresenta uno schema in cui sono presenti 10 righe e 10 colonne, più o meno come nel classico gioco della battaglia navale.

Si può notare che nelle 100 (10 x 10) caselle sono presenti, in prevalenza, i valori nulli. Vi sono anche, evidenziati in reverse, 10 valori unitari ed un numero 2 (terza riga, terza colonna).

Facile è l'analogia della figura con una ipotetica mappa stradale formata da dieci strade orizzontali che si intersecano con altrettante strade verticali; in ogni intersezio-

SYS 49152

Questa, come molti potranno facilmente immaginare, è la SYS di partenza del fumetto: dato questo comando, la storia può iniziare...

Dopo le sue ultime imprese, il nostro eroe Primo Giovedini era stato più volte insignito di alte onorificenze per meriti combattentistici ed informatici.
Tra queste decorazioni spiccavano in particolare ...

```
100 REM ALLA SCOPERTA DELLE ADVENTURES
110 REM PRIMA PARTE: LA MAPPA
120 :
130 DIM A(10,10):REM CREAZIONE DI UNA "SUPERFICIE" BIDIMENSIONALE
140 PRINT CHR$(147):REM CARATTERE SPECIALE PER CANCELLARE SCHERMO
150 CLR: FOR I=1 TO 10:REM CREAZIONE DI DIECI LUOGHI "SPECIALI"
160 GOSUB 310:X=Z       :REM MEDIANTE LA FUNZIONE
170 GOSUB 310:Y=Z       :REM RANDOM (ASCISSA, ORDINATA)
180 A(X,Y)=1: NEXT
190 GOSUB310:X=Z:GOSUB310:Y=Z:A(X,Y)=2:REM DETERMINAZIONE OBIETTIVO
200 FOR I=1 TO 10: REM VISUALIZZAZIONE DELLA MAPPA
210 FOR J=1 TO 10: REM RIGA PER RIGA
220 IF A(I,J)>0 THEN PRINT CHR$(18);:REM REVERSE
230 PRINT A(I,J);CHR$(146);
240 NEXT J: PRINT: PRINT
250 NEXT I
260 PRINT "PREMI UN TASTO":REM MESSAGGIO
270 GET A$:IF A$="" THEN 270:REM ESAME TASTO PREMUTO
280 PRINT CHR$(145) "                    ":REM CANCELLA MESSAGGIO
290 PRINT CHR$(19):GOTO 150:REM RICOMINCIA DALL'INIZIO
300 END
310 Z=INT(RND(0)*11):IF Z=0 THEN 310: REM ESCLUDE COORDINATA ZERO
320 RETURN
```

ne (incrocio) è presente un numero che può assumere il valore 0, 1 oppure 2.

Se, appunto, decidiamo di assegnare al numero 2 il simbolo dell'OBIETTIVO da raggiungere, avremo a disposizione un'efficace mappa da percorrere.

Naturalmente il numero 1 rappresenterà un OSTACOLO insuperabile, mentre i vari "0" potranno essere attraversati senza alcuna difficoltà.

Per semplificare le cose supporremo che il cammino, giunti ai confini Est, Ovest, Nord e Sud, non possa proseguire ulteriormente.

IL PRIMO PROGRAMMA

Il listato n.1 non fa altro che realizzare un percorso del tipo esaminato.

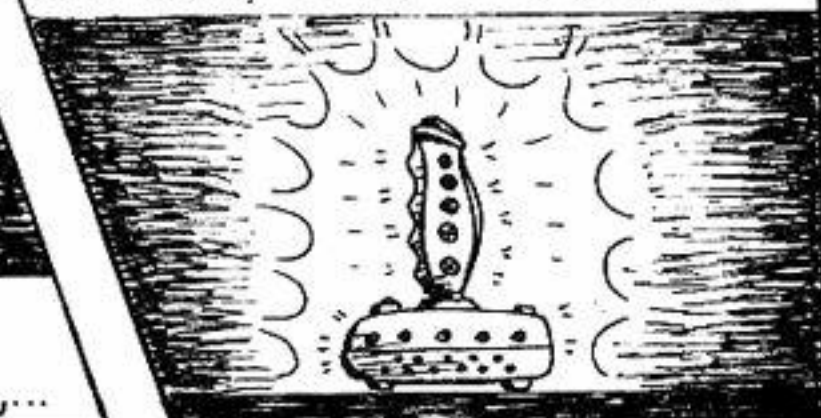
La superficie bidimensionale è simulata efficacemente grazie alla riga 130 che contiene il comando DIMA(10,10).

Questo costringe il computer a riservare spazio in memoria in modo da ospitare 100 numeri, forniti dal prodotto dei numeri 10 x 10, (detti: ARGOMENTI) presenti tra le parentesi del comando stesso.

In effetti lo spazio riservato è maggiore perchè il computer non considera solo 10 righe e 10 colonne, ma 11 e 11 dal momento che il conteggio inizia sempre dal numero zero: si hanno a disposizione, quindi, 11 x 11 (= 121) elementi ma, per semplicità, ne verranno usati solo 100 (numerati da 1 a 10 sia per le righe sia per le colonne).

Dovrebbe esser noto che, non appena una matrice viene dimensionata, ogni suo elemento risulta azzerato. All'inizio, pertanto, si hanno a disposizione 100 caselle contenenti, ciascuna, il valore nullo.

Grazie alla subroutine 310 vengono generati numeri interi casuali, compresi tra 1 e 10, secondo una tecnica descritta molte volte su queste stesse pagine e che quindi

Con un po' di fantasia è possibile inserire ostacoli, premi, nemici ed altri... oggetti

Una matrice bidimensionale simula in modo efficace il labirinto da percorrere

non ripeteremo (un intero inserto sulla funzione RND è stato pubblicato su C.C.C. n. 38).

Le righe 160 e 170 provvedono, servendosi della subroutine 310, ad individuare 10 coordinate casuali X ed Y che, in effetti, rappresenteranno altrettanti elementi della matrice A(X,Y) ai quali (riga 180) verrà assegnato il valore unitario, modificando quello originale che, lo ricordate?, era zero.

L'ultima operazione (riga 190) individua un elemento casuale al quale assegnerà il valore 2.

Le altre righe del programma non fanno altro che visualizzare la matrice-labirinto così realizzata; dopo averla esaminata sarà possibile, premendo un tasto qualunque, generarne un'altra per verificare l'effettiva possibilità di avere a disposizione un numero pressochè infinito di percorsi, di ostacoli e di obiettivi.

IL SECONDO LISTATO

Il problema, ora, è di individuare una valida simulazione che ci consenta di percorrere il labirinto in modo da raggiungere l'obiettivo (2) evitando gli ostacoli (1) ed impedendo lo sconfinamento del labirinto stesso.

Naturalmente, alla partenza, il giocatore non deve sapere il punto in cui si trova ma, tentando di andare nelle quattro direzioni (N, S, E, O), deve sfruttare le "risposte" del computer per orientarsi correttamente.

Il secondo programma ricalca, più o meno (righe 130 - 190) la tecnica usata in precedenza. La riga 200, in effetti, dovrebbe esser cancellata dopo la verifica del corretto funzionamento del programma: visualizza, infatti, le coordinate dell'obiettivo da raggiungere! Analogamente la riga 210, dopo aver determinato casualmente la posizione di partenza, ne visualizza le coordinate; anche in questo caso, per aumentare le difficoltà del gioco, andrebbe rimossa la parte di riga 210 da "Print..." in poi.

Subito dopo (riga 230) un banale controllo del tasto premuto consente di determinare la direzione desiderata che verrà evidenziata dopo la pressione del tasto stesso.

Escludendo il caso (banale) della involontaria pressione di altri tasti diversi da N, S, E, O (riga 280) sarà necessario determinare il comportamento del programma nel caso le direzioni richieste siano lecite.

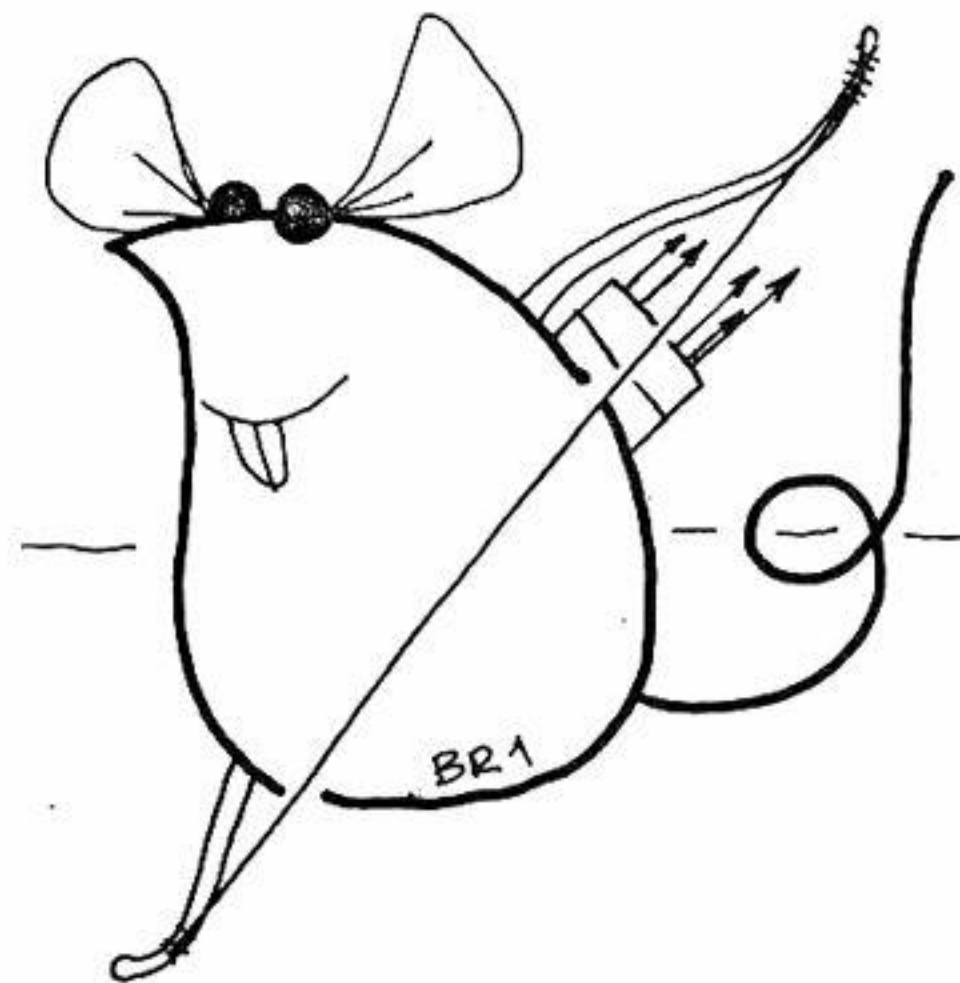
Vi saranno, pertanto, quattro "filtri" software, uno per ciascuna direzione possibile. Ne esamineremo uno solo (righe 440 - 480) perchè i rimanenti sono sostanzialmente simili.

IL "FILTRO"

E' bene sottolineare che la coordinata X aumenta procedendo da sinistra a destra, mentre Y dall'alto in basso.

All'inizio (riga 210) il giocatore si trova in una casella di coordinate X1 ed Y1. Chiedendo di andare a Sud (riga 250) si perviene al filtro-subroutine di riga 450 che, come prima operazione, provvede ad incrementare la sola coordinata Y1. Se, effettuata la somma, tale valore è maggiore di 10 vuol dire che stiamo per uscire dai confini "leciti" del percorso: l'ordine non viene quindi eseguito, la coordinata Y1 ritorna al valore precedente (Y1 = 10) e compare l'opportuno messaggio prima di far ritorno al programma principale.

Nel caso in cui le coordinate siano lecite è necessario verificare se la casella in cui stiamo andando contiene il valore 2: in questo caso, infatti (riga 460) l'obiettivo è stato rag-

```
100 REM ALLA SCOPERTA DELLE ADVENTURES
110 REM SECONDA PARTE: L'ESPLORAZIONE
120 :
130 DIM A(10,10):REM CREAZIONE DELLA SUPERFICIE BIDIMENSIONALE
140 PRINT CHR$(147):REM CARATTERE SPECIALE PER CANCELLARE SCHERMO
150 CLR: FOR I=1 TO 10:REM CREAZIONE DEI DIECI LUOGHI SPECIALI
160 GOSUB 310:X=Z      :REM MEDIANTE LA FUNZIONE
170 GOSUB 310:Y=Z      :REM RANDOM (ASCISSA, ORDINATA)
180 A(X,Y)=1: NEXT
190 GOSUB310:X=Z:GOSUB310:Y=Z:A(X,Y)=2:REM DETERMINAZIONE OBIETTIVO
200 PRINT "OBIETTIVO IN (X,Y):"X;",";Y
210 GOSUB 310: X1=Z: GOSUB 310: Y1=Z: PRINT "SEI IN X="X1;"Y="Y1
220 PRINT "CHE COSA VUOI FARE? (N, S, E, O)";
230 GET A$: IF A$= "" THEN 230
240 IF A$="N"THEN PRINTA$:GOSUB 390:GOTO 220
250 IF A$="S"THEN PRINTA$:GOSUB 450:GOTO 220
260 IF A$="E"THEN PRINTA$:GOSUB 510:GOTO 220
270 IF A$="O"THEN PRINTA$:GOSUB 570:GOTO 220
280 GOTO 230:REM ESCLUDE ALTRI TASTI
290 :
300 REM SUBROUTINE GENERAZIONE COORDINATE RANDOM
310 Z=INT(RND(0)*10):IF Z=0 THEN 310: REM ESCLUDE COORDINATA ZERO
320 RETURN
330 :
340 REM SUBROUTINE VISUALIZZAZIONE MAPPA ADVENTURE
350 PRINTCHR$(147):FOR I=1 TO 10: REM VISUALIZZAZIONE DELLA MAPPA
360 FOR J=1 TO 10: PRINT A(J,I);:NEXT J:PRINT:PRINT: NEXT I: RETURN
370 :
380 REM ESAME SPOSTAMENTO DIREZIONE NORD
390 Y1=Y1-1:IF Y1<1 THEN Y1=1:PRINT"CONFINE NORD!":RETURN
400 IF A(X1,Y1)=2 THEN 630
410 IF A(X1,Y1)=1 THEN PRINT"OSTACOLO!": Y1=Y1+1
420 RETURN
430 :
440 REM ESAME SPOSTAMENTO DIREZIONE SUD
450 Y1=Y1+1:IF Y1>10 THEN Y1=10:PRINT"CONFINE SUD!":RETURN
460 IF A(X1,Y1)=2 THEN 630
470 IF A(X1,Y1)=1 THEN PRINT"OSTACOLO!": Y1=Y1-1
480 RETURN
490 :
500 REM ESAME SPOSTAMENTO DIREZIONE EST
510 X1=X1+1:IF X1>10 THEN X1=10:PRINT"CONFINE EST!": RETURN
520 IF A(X1,Y1)=2 THEN 630
530 IF A(X1,Y1)=1 THEN PRINT"OSTACOLO!": X1=X1-1
540 RETURN
550 :
560 REM ESAME SPOSTAMENTO DIREZIONE OVEST
570 X1=X1-1:IF X1<1 THEN X1=1:PRINT"CONFINE OVEST!":]RETURN
580 IF A(X1,Y1)=2 THEN 630
590 IF A(X1,Y1)=1 THEN PRINT"OSTACOLO!": X1=X1+1
600 RETURN
610 :
620 REM INDIVIDUAZIONE OBIETTIVO E FINE DEL GIOCO
630 PRINT"HAI VINTO!":PRINT"(PREMI UN TASTO)"
640 GET A$: IF A$= "" THEN 640
650 GOSUB 350:END
```

giunto e viene attivata l'opportuna routine (da 630 in poi).

Non basta; bisogna verificare se la nuova casella non contenga un valore unitario. Sappiamo, infatti, che non è possibile superare l'ostacolo; l'operazione viene impedita (riga 470) e la coordinata Y1 torna ad avere il valore precedente (Y1 = Y1 - 1).

In tutti gli altri casi, se cioè la nuova casella contiene valore nullo, il passaggio di casella viene accettato ed il programma chiede una nuova direzione.

TASTO CHIAMA VIDEO

Utilizziamo la tastiera del C/64 in modo diverso dal solito; ricorrendo, ovviamente, al linguaggio macchina

di **Domenico Pavone**

Di Interrupt non si parla mai abbastanza

Dovrebbero ormai essere più che scontati gli enormi vantaggi offerti dalla programmazione in Assembly, ma, giusto per chi "non c'era", ribadiamo ancora una volta i due aspetti più significativi: l'elevata velocità di esecuzione nonchè la possibilità di applicazioni altrimenti impossibili con il pur valido basic.

Avendo già verificato, in precendenza, tali caratteristiche, siamo ora pronti per riassumerle in un'unica realizzazione, passaggio obbligato per chi si accosta al linguaggio macchina (LM): la ridefinizione della tastiera.

Più precisamente, realizzeremo un'utility che, assegnando a tre dei tasti funzione (F1, F3, F5) l'esecuzione di particolari compiti, renderà più comoda la manipolazione di schermate in bassa risoluzione.

Come vedremo meglio tra breve, inoltre, faremo uso della memoria "nascosta" del C/64, lasciando così integro tutto lo spazio riservato al basic e (ad eccezione del centinaio di byte della routine qui proposta) anche quello comunemente usato per allocarvi programmi LM.

Dulcis in fundo, avremo modo di vedere all'opera un importante gruppo di istruzioni assembly, gli operatori logici.

PROVA SU STRADA

Prima di esaminarne la struttura, cominciamo col vedere all'opera la nostra routine: copiate attentamente il listato basic di queste pagine e, dopo averlo opportunamente salvato, date il classico RUN.

Nel caso si sia commesso qualche errore nel digitare le linee DATA, apparirà una segnalazione per indicare il triste avvenimento, altrimenti la SYS di riga 50 attiverà il programma lm che ridefinisce i tasti F1, F3 e F5 in modo che svolgano le loro nuove mansioni.

In particolare, premendo il tasto F1, verrà memorizzato tutto ciò che è visualizzato sullo schermo, mentre, con F5, si potrà, in qualunque momento, richiamare la videata; il tasto F3 si limiterà a porre in reverse l'intero schermo.

Il tutto, ovviamente, in una frazione di secondo e, per ciò che riguarda l'operato di F1 e F5, con modalità ben diverse da quanto già visto sul n.54 della rivista. Ma di questo se ne parlerà tra breve; per adesso limitiamoci a considerare come l'implementazione di queste funzioni non interferisca minimamente con la normale attività del computer (programmazione in basic compresa), e come resti stabile fino a che non si resetti il sistema o si prema Run / Stop e Restore.

Anche in questo caso, comunque, è sufficiente una nuova SYS 49152 per riattivare la routine, ovviamente purchè non si sia spento e riacceso l'apparecchio.

In pratica si è realizzata, con l'aiuto dell'ormai consueto Macro Assembler Commo-

GIA' FATTO

Riportiamo un breve elenco degli argomenti affrontati su C.C.C. per ciò che riguarda i programmi scritti con il Macro Assembler Commodore.

n. 51 - L'assembly, questo sconosciuto

n. 52 - Uno sprite con una marcia in più

n. 54 - Uno schermo di scorta

n. 55 - Un po' d'ordine in memoria

n. 56 - Un antilist a prova di "hacker"

dore, la struttura di base di un semplice editor grafico, le cui prestazioni possono ampiamente essere migliorate: per esempio, dopo l'utilizzazione del tasto F1 si può fare in modo che un messaggio comunichi l'avvenuta memorizzazione della schermata, o ancora, utilizzare il tasto F7 per altre implementazioni (come salvare la schermata su dischetto).

Troppo complicato?

Esaminando a fondo lo sviluppo in Assembly della routine tutto sembrerà più facile. In caso contrario, sarà opportuno rinfrescarsi le idee con qualche ulteriore "spulciatina " ai precedenti numeri della rivista.

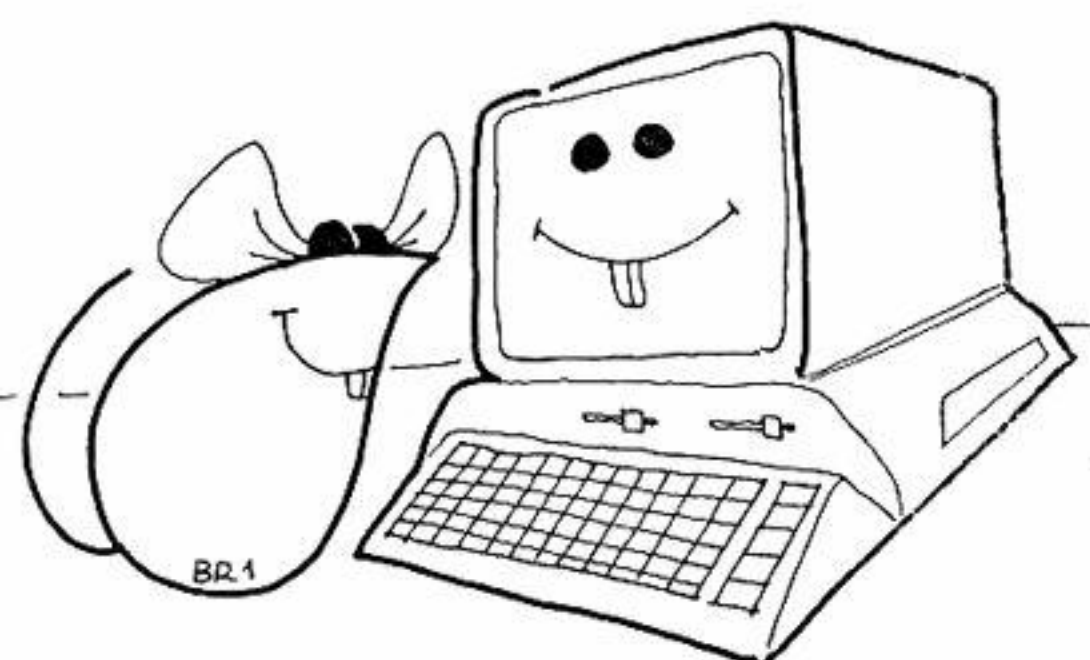
E' PERMESSO INTERROMPERE

Il disassemblato di queste pagine è realizzato (come sempre) con l'editor del MacroAssembler, e quindi leggibile con la stessa facilità di un listato basic.

Allorchè viene richiamato (dalla già menzionata SYS) il programma LM si limita ad eseguire le righe di istruzioni da 9 a 15, la cui unica mansione è quella di modificare il contenuto di due importanti locazioni di memoria, 788 e 789, inserendovi (nel formato basso / alto) l'indirizzo di inizio vero e proprio della routine (label ROUT).

E' questa una procedura già vista in un precedente articolo, allorchè si alterava il vettore di LIST per apportare variazioni. In questo caso, il vettore modificato è quello di INTERRUPT (IRQ), ossia il vettore che normalmente "punta " alla routine di sistema che presiede alla gestione delle interruzioni.

Senza dilungarci sull'argomento (trattatto forse più di ogni altro da Commodore Computer Club), basterà per ora sapere che, così facendo, la routine puntata verrà eseguita ogni sessantesimo di secondo, a patto che la stessa si concluda con un ritorno al normale indirizzo di sistema, $EA31 (riga 70del disassemblato).

Nel modo standard i tasti funzione del C/64 non sono utilizzati. Vi spieghiamo come assegnar loro alcuni compiti da svolgere

I TRUCCHI DEL MESTIERE

Abbiamo più volte constatato come, passando dalla programmazione in basic a quella in Assembly, alcune strutture logiche dei due linguaggi siano molto simili, se non addirittura identiche.

E' questo il caso delle istruzioni di salto, il cui accostamento è abbastanza immediato: al GOTO del basic corrisponde il JMP (jump) in Assembly, mentre lo mnemonico JSR (jump to subroutine) equivale a GOSUB. E fin qui niente di nuovo.

Sappiamo anche che in linguaggio macchina, per creare diramazioni nel flusso del programma (in basic = IF... THEN GOTO...), si ricorre ad una serie di istruzioni di "branch", ovvero di salto condizionato (BEQ, BNE, ecc.).

Spesso però, anche se la cosa può creare problemi di chiarezza per un principiante, quest'ultimo tipo di comandi viene usato al posto dei salti incondizionati, allo scopo di soddisfare due esigenze che assillano perennemente l'incontentabile estro del programmatore in linguaggio macchina:

1) Fare in modo che le routine occupino il minor spazio possibile risparmiando anche il singolo byte, il che si traduce in un ulteriore incremento della già notevole velocità di esecuzione.

2) Rendere il programma facilmente "trasportabile".

Per essere più chiari, riferiamoci al disassemblato della routine proposta in queste pagine, e più precisamente all'istruzione presente in riga 46 (lo stesso vale anche per riga 62). Il suo significato è abbastanza chiaro: se Y è uguale a zero, salta alla riga 56. Tuttavia, l'istruzione precedente è anch'essa un salto condizionato, per cui sarebbe altrettanto corretto usare JMP al posto di BEQ, come l'abitudine al basic ci consiglierebbe.

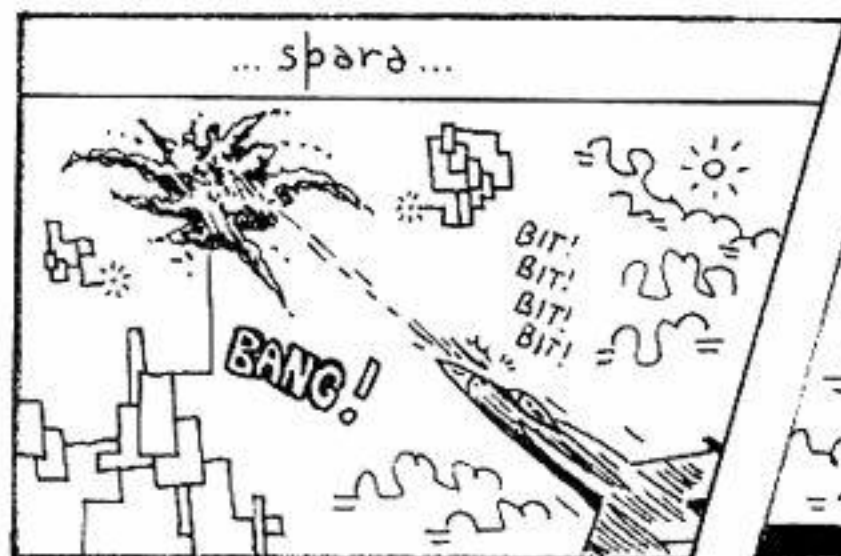
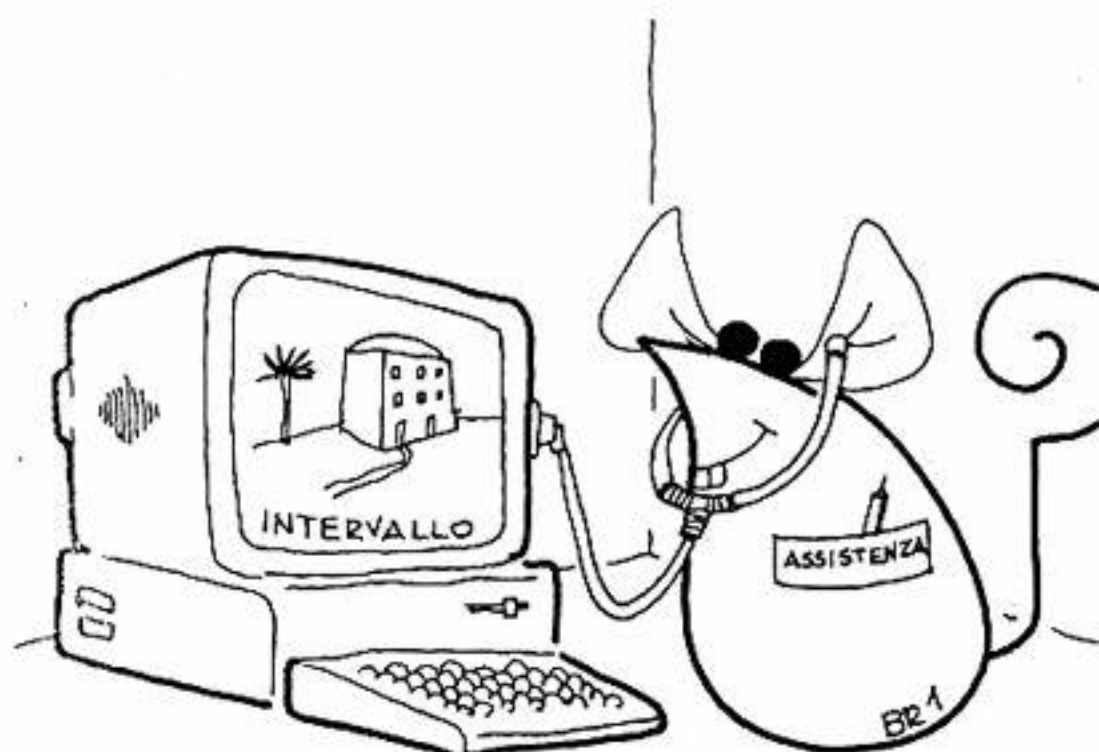
Vediamo invece cosa ci mostra il nostro beneamato Macro

Per modificare il vettore occorrerà dapprima escludere le interruzioni; ciò è realizzato dall'istruzione assembly SEI (SEt Interrupt), la quale setterà (cioè porrà ad 1) il bit 2 del registro di stato (vedi n.55). Con CLI (CLear Interrupt) si azzererà lo stesso bit, ripristinando il normale ciclo di interruzioni.

Che cosa avverrà ogni sessantesimo di secondo?

Semplicemente, le istruzioni contenute nelle righe da 17 a 24 controlleranno quale tasto è stato premuto, continuando l'esecuzione della routine di queste pagine solo se si tratta di uno dei tre tasti funzione interessati. In caso contrario il controllo viene restituito al sistema operativo.

E qui apriamo una breve parentesi.

Esistono varie tecniche per esaminare

Assembler, o meglio il suo monitor LM.
Caricate in memoria la routine pubblicata in queste pagine, dopodichè, col dischetto Macro Assembler debitamente inserito nel drive, impartite...
Load "Monitor$8000",8,1
...e, a caricamento ultimato:
SYS 32768
A questo punto, digitate:
D C040 C069 [R]
Apparirà la sezione finale del disassemblato della routine.
L'istruzione BEQ che stiamo considerando è quella allocata in C042 e, come si può notare, è formata da due byte (F0 e 0D). Se, invece, avessimo usato JMP, pur ottenendo lo stesso risultato, l'occupazione di memoria sarebbe stata di tre byte, come è possibile verificare esaminando l'ultima istruzione sullo schermo
C069 4C 31 EA JMP $EA31
Proviamo ora a spostare la routine in un'altra zona di memoria, operazione che il monitor del MacroAsssembler consente di eseguire con estrema facilità: basta infatti digitare...
T C000 C06B 2000
..cioè: trasferisci le locazioni da $C000 a $C06B nelle locazioni da $2000 in poi, ed il gioco è fatto.
Se ora impartiamo D 2040 2069, rivedremo la stessa sezione della routine in esame, ma con gli operandi di tutte le istruzioni di salto condizionato aggiornati alla nuova allocazione. Il che, tradotto in parole povere, significa risparmiare una non indefferente mole di lavoro.
Lo stesso non avviene per i salti incondizionati JMP e JSR, ed ecco spiegata la propensione all'uso, nell'ambito di una stessa routine, dei comandi di "branch".

l'input da tastiera, per lo più basate sul buffer di tastiera, o su particolari routine del Kernal. Quella adottata è una delle più semplici e veloci, ma richiede qualche chiarificazione.

Come è possibile rilevare dal disassemblato (riga 17), si fa ricorso alla locazione $CB (decimale 203), nella quale il sistema deposita il codice del tasto premuto. Si tratta però di valori particolari, utilizzati dal S.O. per decodificare la tastiera, e quindi non corrispondenti nè allo standard ASCII nè ai codici di schermo.

Ai fini della nostra routine è sufficiente sapere che, ai tasti funzione F1, F3 ed F5, corrispondono i valori 4, 5 e 6, ma se volete conoscerne altri, basterà premere i tasti desiderati dopo aver lanciato il seguente mini-

```
                OPERAZIONI LOGICHE

Tavola verita'        Esempi in Assembly

 1 AND 1 = 1          ACC. 10011101   dec. 157
 1 AND Ø = Ø          AND  11111000   dec. 248
 Ø AND 1 = Ø          -------------
 Ø AND Ø = Ø          ACC. 10011000   dec. 152
-----------------------------------------------
 1 OR  1 = 1          ACC. 10011101   dec. 157
 1 OR  Ø = 1          ORA  00000111   dec.   7
 Ø OR  1 = 1          -------------
 Ø OR  Ø = Ø          ACC. 10011111   dec. 159
-----------------------------------------------
 1 EOR 1 = Ø          ACC. 01000001   dec.  65
 1 EOR Ø = 1          EOR  10000000   dec. 128
 Ø EOR 1 = 1          -------------
 Ø EOR Ø = Ø          ACC. 11000001   dec. 193
```

Il disassemblato aiuterà a comprendere il funzionamento della routine proposta

programma basic...

10 Print chr$(147); Peek(203): Goto10

...tenendo presente che il valore 64 corrisponde a nessun tasto premuto.

ECCESSO DI VELOCITA'

Un altro aspetto da chiarire riguarda le istruzioni alle righe 18-20:

Il codice del tasto premuto è depositato nel registro X, ma, prima di verificare se è compreso tra 4 e 6, lo si confronta con il contenuto della locazione $FA (decimale 250), dove la riga 20 lo deposita, uscendo in caso di eguaglianza. A quale scopo?

Beh, il fatto è che il linguaggio macchina, paradossalmente, presenta un inconveniente: è TROPPO veloce. Senza queste istruzioni, infatti, a meno di addestrare le nostre dita a tocchi da pianista, la pressione dei tasti funzione risulterebbe, nella maggior parte dei casi, troppo prolungata, attivando più volte le routine interessate.

Nel caso di F3, ad esempio, lo schermo verrebbe posto in reverse un imprecisato numero di volte, magari tornando al suo stato di partenza, creando così l'illusione che

```
                    D I S A S S E M B L A T O
---------------------------------------------------------------------------

01 IRQ      =$314         ;dec.788
02 PUNT1    =$FB          ;dec.251
03 PUNT2    =$FD          ;dec.253
04 REG      =$01
05 AREA1    =$400         ;dec.1024
06 AREA2    =$A000        ;dec.40960
07         *=$C000        ;dec.49152
08 ;--------------------------------
09          SEI           ;Dirotta
10          LDA #<ROUT    ;vettore di
11          LDX #>ROUT    ;interrupt
12          STA IRQ       ;all'inizio
13          STX IRQ+1     ;della
14          CLI           ;nostra
15          RTS           ;routine.
16 ;--------------------------------
17 ROUT     LDX $CB       ;Legge codice
18          CPX $FA       ;tasto premuto
19          BEQ USCITA    ;ed esce se
20          STX $FA       ;uguale a se
21          CPX #4        ;stesso, o se
22          BMI USCITA    ;diverso dai
23          CPX #7        ;tasti funzio-
24          BPL USCITA    ;ne F1/F3/F5.
25 ;--------------------------------
26          LDY #00       ;Prepara due
27          STY PUNT1     ;puntatori in
28          STY PUNT2     ;pagina 0 con
29          LDA #>AREA1   ;indirizzo di
30          STA PUNT1+1   ;partenza e
31          STA $02       ;arrivo, piu'
32          LDA #>AREA2   ;loc.2 come
33          STA PUNT2+1   ;contatore.
34 ;--------------------------------
35          CPX #6        ;Se tasto F5,
36          BEQ F5              ;salta.
37 F1F3     LDA (PUNT1),Y       ;Loc. partenza.
38          CPX #5              ;Se <> da F3,
39          BNE STORE           ;salta.
40          EOR #128            ;Inverte char.
41          STA (PUNT1),Y       ;e lo deposita.
42          JMP CONT            ;Salta.
43 STORE    STA (PUNT2),Y       ;Deposita in
44 CONT     INY                 ;area 2.
45          BNE F1F3            ;Se Y<>0 salta.
46          BEQ INCREM          ;Se Y=0 salta.
47 ;--------------------------------------
48 F5       LDA REG             ;Azzera bit 0
49          AND #254            ;del registro
50          STA REG             ;I/O in loc.1.
51 LOAD     LDA (PUNT2),Y       ;Trasferisce
52          STA (PUNT1),Y       ;da RAM sotto
53          INY                 ;il basic ad
54          BNE LOAD            ;area video.
55 ;--------------------------------------
56 INCREM   INC PUNT1+1         ;Incrementa
57          INC PUNT2+1         ;byte alto
58          DEC $02             ;dei puntatori
59          BEQ FINE            ;4 volte.
60          CPX #6              ;Ritorna al
61          BNE F1F3            ;trasferimento
62          BEQ LOAD            ;in corso.
63 ;--------------------------------------
64 FINE     CPX #6              ;Se tasto <> F5
65          BNE USCITA          ;esce,
66          LDA REG             ;altrimenti
67          ORA #1              ;setta ad 1 il
68          STA REG             ;bit 0 di loc.1
69 ;--------------------------------------
70 USCITA JMP $EA31             ;IRQ del S.O.
```

non sia avvenuto niente (tra imprecazioni seguite dal classico: "non funziona!! ").

MEMORIA CON DOPPIO FONDO

Entriamo ora nel "cuore" della routine, vagliando le istruzioni attivate a seconda del valore del registro X, che manterrà sempre inalterato il suo contenuto, ossia il codice del tasto funzione premuto.

Con le righe da 26 a 33 si preparano due puntatori in pagina zero ($FB, $FC e $FD, $FE) contenenti uno l'indirizzo di partenza dell'area video (1024) e l'altro la prima lo-

```
10 REM ---------------------------------------------------
15 REM            RIDEFINIZIONE TASTI F1 - F3 - F5
20 REM ---------------------------------------------------
25 :
30 FORX=0TO108:READA:B=B+A:POKE49152+X,A:NEXT
35 IFB<>15061THENPRINT"ERRORE NELLE LINEE DATA!":END
40 PRINT"TASTO F1 = MEMORIZZA SCHERMATA"
45 PRINT"TASTO F3 = REVERSE DEL VIDEO"
50 PRINT"TASTO F5 = MOSTRA SCHERMO MEMORIZZATO":SYS49152
55 :
60 DATA 120,169,013,162,192,141,020,003,142,021,003,088,096,166
65 DATA 203,228,250,240,087,134,250,224,004,048,081,224,007,016
70 DATA 077,160,000,132,251,132,253,169,004,133,252,133,002,169
75 DATA 160,133,254,224,006,240,020,177,251,224,005,208,007,073
80 DATA 128,145,251,076,064,192,145,253,200,208,238,240,013,165
85 DATA 001,041,254,133,001,177,253,145,251,200,208,249,230,252
90 DATA 230,254,198,002,240,006,224,006,208,211,240,235,224,006
95 DATA 208,006,165,001,009,001,133,001,076,049,234
96 END
```

Memorizzare una schermata, richiamarla, o porla in reverse, non è un più un problema con il l.m.

cazione normalmente occupata dall'interprete del basic (40960).

I puntatori serviranno tanto per il trasferimento, in un senso o nell'altro, dell'area video, che per il reverse dello schermo, in quanto tutte e tre le opzioni sfruttano, per operare, l'indirizzamento indiretto postindicizzato (righe 37, 41, 43, 51, 52), già trattato in precednza sulla rivista.

Come già accennato, una peculiarità della routine consiste nel fatto che le schermate vengono memorizzate, lontane da occhi indiscreti, nell'area RAM posta "sotto" l'interprete basic, e della quale potete avere maggiori notizie ripescando il n.53 della rivista (pag.71).

Sappiamo che per immettere dati in queste locazioni non c'è alcun problema: basta "pokarvi" (per gli appassinati di LM = "storarvi") qualcosa, che immancabilmente finirà in RAM.

Per leggerli, però, la procedura risulta più complessa.

In sintesi, per accedere agli 8 Kb, numerati da 40960 a 49151, occorre modificare lo stato del bit 0 del registro di Input / Output del nostro amato computer, ossia la locazione 1.

Se questo bit è posto a 1 (settato), nelle suddette locazioni sarà presente la ROM basic, se invece lo stesso bit è posto a 0 (resettato), "emergerà" la RAM.

Mentre per chi usa il basic escludere l'interprete presenta difficoltà (nella maggior parte dei casi, insormontabili), per il programmatore LM quest'area è una vera e propria manna, in quanto utilizzabile con estrema facilità.

A patto che si sappia come modificare lo stato di un singolo bit di una locazione.

UNA LOGICA DI FERRO

Ed ecco che entrano in gioco gli operatori logici, ossia le istruzioni assembly AND, ORA ed EOR, che potete ammirare, nella nostra routine, alle righe 40, 49 e 67. Tutti e tre agiscono esclusivamente sul contenuto del registro accumulatore, nel quale viene poi depositato l'esito dell'operazione (ovviamente il precedente contenuto di A viene perso).

Alla stessa categoria appartiene anche lo mnemonico BIT, del quale però ci occuperemo in future applicazioni.

Tornando in argomento, gli operatori AND e ORA sono proprio gli strumenti che, in linguaggio macchina, consentono, ri-

Sfruttiamo la possibilità di utilizzare la Ram "nascosta" sotto il Basic del C/64

spettivamente, di azzerare e settare i singoli bit di un byte. Vediamo di capire come.

Con AND si effettua l'omonima operazione tra l'accumulatore e l'operando dell'istruzione. Riferendoci alla nostra routine (righe 48 - 50), in A è presente il contenuto della locazione 1, e quindi si effettuerà l'AND di tale valore con 254, ossia con il suo corrispettivo binario 11111110.

Il motivo è presto detto: se osservate il riquadro contenente le cosiddette "tabelle della verità" (Mosè non c'entra!), potete rilevare come il risultato di un AND sarà 1 solo se entrambi i bit confrontati sono uguali ad 1.

Poichè l'istruzione AND effettua l'operazione su tutti i bit del contenuto dell'accumulatore, ecco che adoperando il valore binario 11111110 (detto "maschera"), si avrà come risultato un numero binario in cui il bit 0 sarà in ogni caso 0, mentre gli altri rimarranno invariati.

Quindi, riassumendo, ad ogni bit posto a zero nella "maschera", corrisponderà, dopo una operazione AND, un bit azzerato, mentre nella stessa posizione dei bit posti ad 1 non si avrà alcun cambiamento.

Analogamente, con ORA (OR Accumulator) si effettua l'OR del contenuto dell'accumulatore con l'operando dell'istruzione assembly. E poichè l'operazione OR dà come risultato 1 solo se almeno uno dei bit è uguale ad uno (vedi riquadro), si produrrà un valore binario in cui ad ogni bit settato nella maschera corrisponderà un bit posto ad uno, mentre gli altri rimarranno invariati.

Ecco dunque che con le istruzioni alle righe 48 - 50 si azzera il bit 0 della locazione 1 (il registro I/O), consentendo l'accesso in lettura alla RAM "nascosta" sotto l'interprete del basic, mentre alle linee 66 - 68 si ripristina lo stato di default del bit (settato) effettuando un OR con il valore binario 00000001.

Leggermente più complicato risulta l'operatore EOR (Exclusive OR = OR esclusivo), il cui comportamento è rilevabile dal solito riquadro. Come potete notare, l'operazione dà 1 se soltanto uno dei due bit confrontati è uguale ad 1. Dal punto di vista pratico questa peculiarità viene spesso sfruttata per due caratteristiche:

1) Effettuando un EOR con 255 (binario 11111111) si ottiene sempre il cosiddetto "complemento" del valore originale, ossia ogni bit 0 risulterà 1, e viceversa.

2) Ripetendo due volte un EOR con lo stesso numero, si torna al valore originale.

E' quest'ultima proprietà dell'operatore che viene sfruttata dalla nostra routine per porre in reverse lo schermo (righe 40 - 41): poichè i codici di schermo dei caratteri in reverse corrispondo a quelli normali addizionati al valore (costante) 128, ecco che basterà prelevare il contenuto della memoria video, effettuare un EOR con 128 (binario 10000000), e quindi ridepositare il risultato nelle stesse locazioni.

Un successivo EOR, e quindi una seconda pressione del tasto F3, riporterà lo schermo ai suoi normali contenuti.

Sulla base di quanto esposto, dovrebbe ora risultare più facile esaminare il disassemblato dell'utility proposta, aiutandosi anche con i commenti posti a lato delle istruzioni.

Tenete presente, inoltre, che con il Macro Assembler in fase di editing è possibile usare qualsiasi notazione numerica: per cui, se la cosa può risultarvi più facile, gli operandi delle istruzioni AND, ORA ed EOR possono essere espressi direttamente in binario, basta farli precedere dal simbolo di percentuale (%).

Concludiamo con una poco ortodossa istruzione in formato MacroAssembler:
.BYTE 'AL PROSSIMO NUMERO!'

RAM, COME E DOVE

Guida pratica alla RAM libera extra Basic del C/16 e del Plus/4

di **Alessandro Diano**

Molte locazioni RAM del C/16 rischiano di restare inutilizzate

Quest'articolo è un'esortazione alla sperimentazione, un osare nella prova che, per chi smanetta su una tastiera, dovrebbe essere la parola d'ordine. E' questo, infatti, l'ostacolo maggiormente impegnativo da superare per imparare un qualsiasi linguaggio di programmazione.

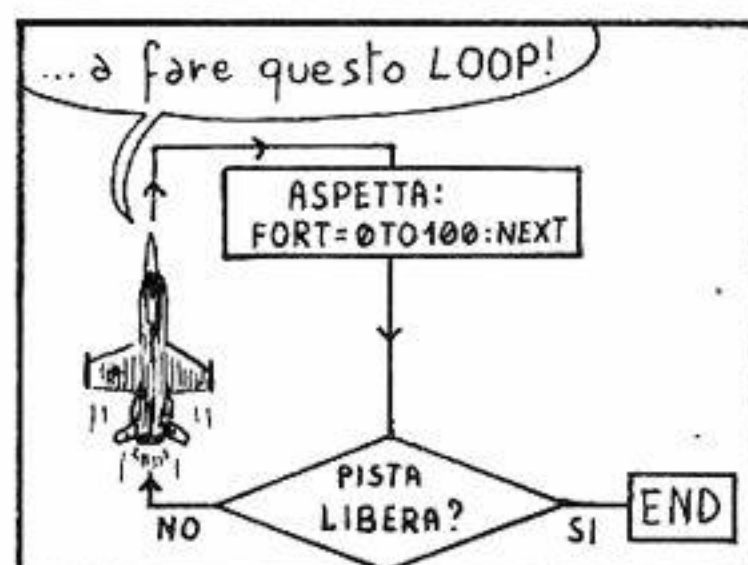
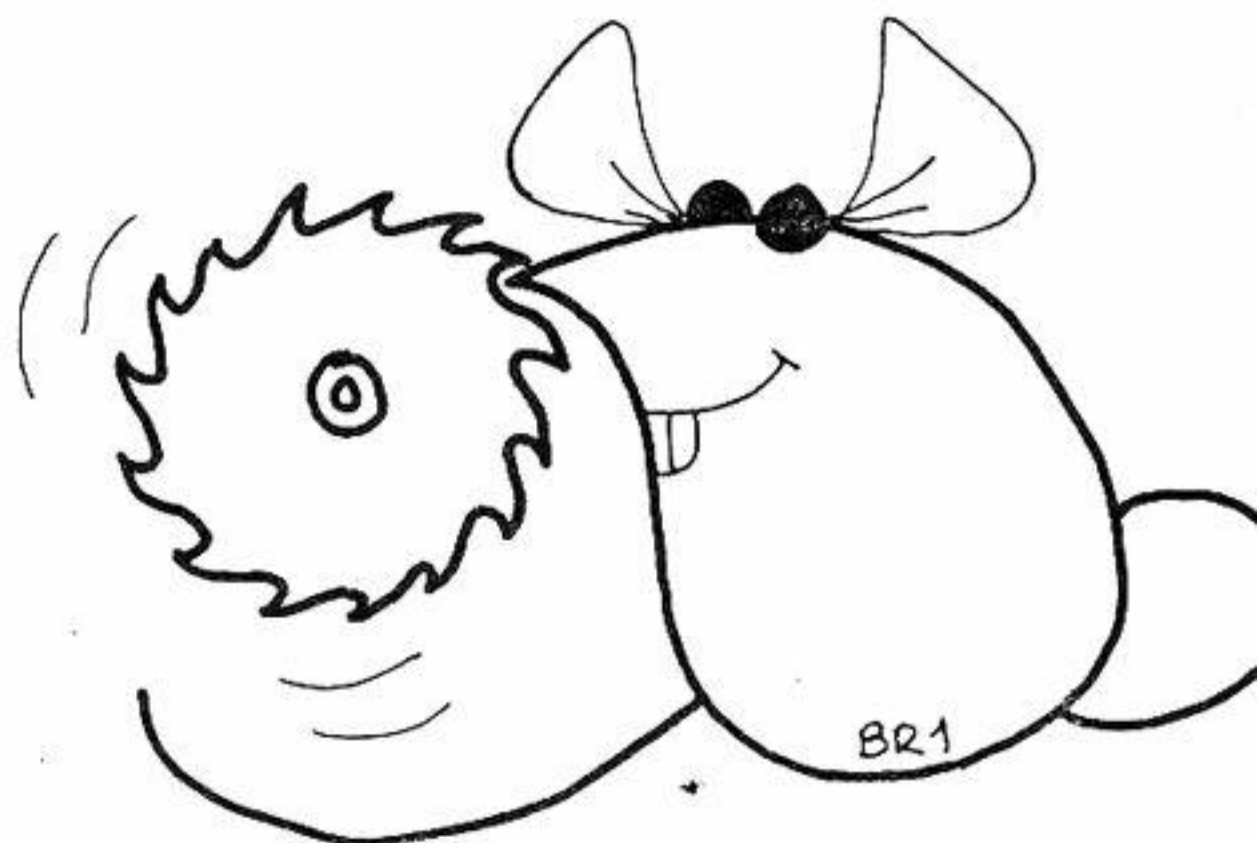
Il tentativo, in altre parole, indica (dopo gli inevitabili sbagli) la strada da seguire in seguito ad un errore logico, di programmazione o di ignor... disinformazione; rappresenta l'unico mezzo per passare dalla "non conoscenza" al "sapere-come" (know-how) per quanto riguarda un qualsiasi problema inerente la creazione di una routine in un qualsiasi linguaggio.

La documentazione, in tale fase, è fondamentale (non si può stilare un decente programma in assembly se non si ha a disposizione una mappa della memoria!) ma l'elemento maggiormente importante è sicuramente rappresentato dalla pazienza relativa alla gestione degli errori incontrati. E tale condizione può solamente essere raggiunta passando attraverso una sperimentazione continua, anche di ciò che sembra assurdo: senza questa mentalità "illegale", infatti, molto difficilmente l'utente normale avrebbe avuto la possibilità di vedere effetti decisamente insoliti in un piccolo computer.

DA PARTE NOSTRA

E' indubbio che l'invito a sperimentare vi sia giunto più volte. "Chi non digita non rosica" recita spesso l'esploratore dell'informatica che passa pomeriggi piovosi durante i quali si introducono istruzioni sconnesse che, "per caso", formano un miniprogramma abbastanza simpatico e, forse, utile.

Qualche tempo fa, ad esempio, mi trovai a considerare l'opportunità di un turbo-tape per C/16 che consentisse di ridurre sensibilmente i tempi d'attesa dovuti al SAVE ma, soprattutto, al LOAD dei programmi da nastro; Commodore Computer Club ne pubblicò, nel frattempo, ben due: il primo, nel numero 42, ad opera dell'oramai leggendario nonchè mitico Pastorelli che però richiedeva il caricamento del programma T/T sia per il SAVE (ovviamente) che per il LOAD; il secondo, tre numeri più tardi, leggermente più lento ma con il vantaggio di registrare insieme ad ogni programma salvato un caricatore che non richiedeva la presenza del programma di turbo-tape durante il LOAD.

```
. 2000  A9 0E     LDA #$0E
. 2002  A2 0F     LDX #$0F
. 2004  A0 20     LDY #$20
. 2006  8D 06 FF  STA $FF06
. 2009  8E FC FF  STX $FFFC
. 200C  8C FD FF  STY $FFFD
. 200F  78        SEI
. 2010  8D 3F FF  STA $FF3F
. 2013  EE 19 FF  INC $FF19
. 2016  4C 00 20  JMP $2000

Disassemblato n. 1
```

```
2000 LDA #$0B   Valore per disabilitare
                lo schermo.
2002 LDX #$0F   Nuovo indirizzo di reset
2004 LDY #$20   $200F nella forma Lo/Hi.

2006 STA $FF06 Disabilita lo schermo.

2009 STX $FFFC Mette il nuovo indirizzo
200C STY $FFFD di reset nelle locazioni
               relative.
200F SEI       Disabilita l'interrupt.
2010 STA $FF3F Disabilita la ROM.
2013 INC $FF19 Incrementa il colore
               della cornice.
2016 JMP $2000 Vai a $2000: loop
               infinito.

Disassemblato n. 2
```

Entrambi, però, presentavano il difetto di occupare parte dello spazio RAM dedicato al Basic, posizionandosi al termine dello stesso: diventava in tal modo impossibile la registrazione di un programma di 12 Kbyte che, come noto, in un C/16 riempie completamente la RAM Basic, oppure, peggio ancora, quella di un listato per Plus/4 ancora più lungo.

Ha avuto così inizio l'esplorazione della memoria antecedente l'indirizzo 4097 ($1001) di inizio Basic, alla ricerca delle zone dove "spezzare" lunghe routine l.m. da realizzare, magari, più per didattica che per effettiva utilità, a causa del disuso nel quale, intanto, cadevano i poveri C/16 e Plus/4.

Qui di seguito sono ben lieto di pubblicare i risultati della ricerca per far conoscere dov'è possibile il posizionamento di routine alternative d'interrupt (tipo orologi sul video et similia) o, più semplicemente, di "Basic tools" che non tolgano parte della già esigua (sul C/16) memoria dedicata ai listati.

RAM PRE-BASIC PER C/16 & PLUS/4

Avere a disposizione una mappa della RAM libera è più utile di quanto possa sembrare

Ecco, qui di seguito, la descrizione della zona RAM più interessante per approfondire gli studi accennati in precedenza:

291 ($124) / 463 ($1CF):
Sono 172 byte nella zona dello stack riservato al sistema operativo; oltre $1CF è prudente non allocare nulla, pena il rischio di sovrapposizioni da parte dei valori immessi nello stack. Questa zona, inoltre, è immune al tasto di reset: i valori che verranno immessi non risentiranno di una sua eventuale pressione.

641 ($281) / 744 ($2E8):
Sono i 104 byte dell'area d'interfacciamento Basic / DOS e della zona utilizzata dai parametri dei comandi grafici; è intuibile che, usando quest'area, sono da evitare sia i comandi per l'alta risoluzione (da GRAPHIC in poi) che un colloquio con l'unità a dischi.

QUANDO SI PREME UN TASTO SUL C/16 & PLUS/4

Vengono elencati i codici (non riportati da alcun tipo di documentazione) che, letti con una PEEK Basic od un Load Assembly dalla locazione denominata SFDX (198 = $C6) oppure dalla LSTX (2038 = $7F6), consentono all'utente di rilevare, via software, la pressione di un qualsiasi tasto della tastiera del computer.

Per quanto riguarda i tasti di Shift, Commodore e Control, è possibile anche la PEEK della locazione SHFLAG posta a 1347 ($543) che conterrà, a seconda del tasto premuto, il valore 0 (nessuno), 1 (Shift), 2 (Commodore), 4 (Control). Se la pressione dei tasti suddetti avviene in contemporanea, il valore letto sarà la somma dei valori relativi ai tasti; il numero 7 (= 1 + 2 + 4), ad esempio, indica che sia lo Shift che il Commodore che il Control sono premuti.

C/16: VALORE LETTO

Decim.	Esad.	TASTO PREMUTO
0	$00	Delete
1	$01	Return
2	$02	£
3	$03	Help/F7
4	$04	F1/F4
5	$05	F2/F5
6	$06	F3/F6
7	$07	Chiocciolina
8	$08	3
9	$09	W
10	$0A	A
11	$0B	4
12	$0C	Z
13	$0D	S
14	$0E	E
15	$0F	Shift
16	$10	5
17	$11	R
18	$12	D
19	$13	6
20	$14	C
21	$15	F
22	$16	T
23	$17	X
24	$18	7
25	$19	Y
26	$1A	G
27	$1B	8
28	$1C	B
29	$1D	H
30	$1E	U
31	$1F	V
32	$20	9
33	$21	I
34	$22	J
35	$23	0
36	$24	M
37	$25	K
38	$26	O
39	$27	N
40	$28	Crsr giu'
41	$29	P
42	$2A	L
43	$2B	Crsr su
44	$2C	.
45	$2D	:
46	$2E	-
47	$2F	,
48	$30	Crsr sinis.
49	$31	*
50	$32	;
51	$33	Crsr destra
52	$34	Escape
53	$35	=
54	$36	+
55	$37	/
56	$38	1
57	$39	Home
58	$3A	Control
59	$3B	2
60	$3C	Spazio
61	$3D	Commodore
62	$3E	Q
63	$3F	Run/Stop
64	$40	Nessun tasto

PER CONOSCERCI MEGLIO

E' nostra consuetudine effettuare, di tanto in tanto, una ricerca per esaminare le reali aspettative di chi acquista la nostra pubblicazione. Il questionario va compilato nel modo più semplice possibile: per ciascuna domanda sono indicate quattro risposte; è sufficiente che il lettore indichi quella scelta con una crocetta nel corrispondente quadratino.

Si ricorda che è indispensabile indicare SOLTANTO una delle quattro risposte. Nel caso in cui nessuna delle risposte indicate vi soddisfi, tracciate la X in quella che più si "avvicina" alla vostra scelta ideale.

Domanda N.1/X

Quali computer possiedi?

- ☐ A Il Commodore 64 oppure il C/128.
- ☐ B Amiga (specificare modello).
- ☐ C Uno dei seguenti computer: Vic/20, C/16, Plus/4.
- ☐ D Nessuno di quelli indicati.

Domanda N.1/Y

Quali computer possiedi?

- ☐ A Un modello Ms-Dos compatibile (specificare quale).
- ☐ B Amiga (specificare modello) OLTRE a C/64 e/o C/128.
- ☐ C Molti computer (specificare).
- ☐ D Non posseggo alcun computer.

Domanda N.2

Quali computer utilizzi di solito (per lavoro, hobby, studio, eccetera)?

☐ A Quello/i indicati nelle risposte precedenti.
☐ B Un altro computer (in ufficio, a scuola).
☐ C Più computer diversi tra loro.
☐ D Un computer di proprietà di altre persone (amico, parente, club, eccetera).

Domanda N.3

Di quali periferiche AGGIUNTIVE (escluse, cioè, quelle standard) è dotato il tuo computer (oppure, se non possiedi un calcolatore: quali periferiche acquisteresti dopo aver acquistato il computer)?

☐ A Solo il registratore.
☐ B Solo il drive (oltre all'eventuale registratore).
☐ C Solo la stampante (oltre all'eventuale registratore).
☐ D Il drive e la stampante.

Domanda N.4

Quale periferica compreresti subito, se te ne capitasse l'occasione?

☐ A Il drive.
☐ B La stampante.
☐ C Una tavoletta grafica oppure altri accessori (digitalizzatore, plotter, modem, mouse).
☐ D Un monitor a colori di qualità.

Domanda N.5

Per quale motivo non hai ancora acquistato il computer oppure la periferica indicata nella domanda precedente?

☐ A Perchè non dispongo della cifra necessaria.
☐ B Perchè aspetto l'occasione di acquistare un computer più potente (Ms/Dos, Amiga o altri).
☐ C Perchè non mi sono strettamente necessari.
☐ D Perchè in casa mia non ho lo spazio necessario per sistemare adeguatamente tanti oggetti.

Domanda N.6

Escludendo il tempo che ti assorbe il lavoro, lo studio oppure la famiglia, quante ore ti restano mediamente (ad esclusione dei periodi di vacanza) da dedicare al tuo tempo libero nei periodi autunno - inverno - primavera (dal lunedì alla domenica inclusi)?

☐ A Oltre 30 ore alla settimana
☐ B Fino a 20 ore " "
☐ C Fino a 10 ore " "
☐ D Meno di 10 ore " "

Domanda N.7

Come utilizzi prevalentemente il computer?

☐ A Per giocare
☐ B Per applicazioni grafiche e musicali
☐ C Per applicazioni professionali
☐ D Per varie applicazioni generali

Domanda N.8

Ti piace applicarti per conoscere i "segreti" del tuo computer?

☐ A Sì, soprattutto studiando e digitando.
☐ B No, preferisco utilizzare software già "pronto" su supporto magnetico.
☐ C Lo farei disponendo di software già pronto su disco, ma da modificare a piacimento.
☐ D Preferisco articoli di didattica che non richiedano digitazione di listati.

Domanda N.9

Riferendoci al linguaggio Basic, la tua esperienza accumulata utilizzando il computer e leggendo libri e riviste ti consente di definirti:

☐ A Appena capace di comprendere gli articoli più semplici pubblicati su Commodore Computer Club.
☐ B In grado di assimilare gli articoli un po' più "pesanti".
☐ C Sufficientemente abile da capire tutti gli articoli più impegnativi.
☐ D Tanto abile da "snobbare" gli articoli che riguardano il Basic e leggere solo altri argomenti (L.M., hardware, novità su periferiche, eccetera).

Domanda N.10

Riferendoti al Linguaggio Macchina (o Assembler) la tua esperienza accumulata utilizzando il computer e leggendo libri e riviste ti consente di definirti:

☐ A Praticamente incapace a capirci qualcosa.
☐ B Sufficientemente abile a seguire un articolo, purchè l'argomento (o il programma) sia breve e semplice.
☐ C In grado di comprendere i vari argomenti affrontati.
☐ D Molto abile, tanto che, degli articoli pubblicati, mi interessano solo i concetti dato che preferisco sviluppare le applicazioni per conto mio.

Domanda N.11

Considerando vera l'affermazione per cui "non si finisce mai di imparare", ti piacerebbe avere una cultura... mostruosa (o comunque più vasta di quella che già possiedi) nel campo dell'informatica?

☐ A No, perchè richiederebbe tempo, voglia e pazienza che non ho oppure perchè, non appena si impara qualcosa, ci si accorge che c'è sempre tanto altro da imparare e non la si finisce più...
☐ B No, perchè l'eventuale esperienza che accumulerei non sarebbe più valida per i computer che verranno prodotti tra pochi anni.
☐ C Sì, ma solo se gli argomenti da affrontare sono trattati in modo semplice ed elementare.
☐ D Sì, purchè non sia una cultura fine a se stessa ma finalizzata alla soluzione di problemi pratici (software professionale, grafica, giochi "evoluti", intelligenza artificiale, eccetera).

Domanda N.12

QUANTI programmi digiti dalle riviste di informatica che compri?

☐ A Quasi tutti quelli pubblicati.
☐ B Al massimo la metà di quelli pubblicati.
☐ C Tutti quelli molto brevi e semplici da digitare.
☐ D Quasi nessuno.

Domanda N.13

QUALI programmi digiti dalle riviste che compri?

☐ A Soprattutto quelli molto brevi, indipendentemente dalla loro utilità.
☐ B Tutti quelli che mi interessano, indipendentemente dalla loro lunghezza.
☐ C Quasi nessuno, ma prendo nota di quelli che potranno servirmi in futuro.
☐ D Quasi nessuno, ma leggendo gli articoli che ne descrivono il funzionamento cerco di realizzare un programma originale che raggiunga lo stesso scopo.

Domanda N.14

Apporti modifiche ai programmi pubblicati sulle riviste?

☐ A No, quasi mai: li digito e li conservo così come sono.
☐ B A volte aggiungo qualche routine grafica, sonora e comunque modifico (di poco) la struttura originale del listato.
☐ C Inserisco variazioni, ma solo una alla volta e solo dopo aver costatato il corretto funzionamento del programma pubblicato.
☐ D Digito direttamente una versione da me stesso modificata, spesso dopo aver dato un semplice sguardo al programma pubblicato.

Domanda N.15

Per ciò che riguarda la nostra rivista, come giudichi Commodore Computer Club?

☐ A Va bene così com'è sia per la varietà di argomenti sia per come li affronta.
☐ B Affronta argomenti troppo spesso complessi e di difficile comprensione; dovrebbe diventare più semplice.
☐ C Affronta argomenti troppo semplici e banali; dovrebbe "salire" di livello.
☐ D Dovrebbe offrire programmi più complessi ma, contemporaneamente, l'opportunità di trovare in edicola il software pubblicato su nastro e/o su disco.

Domanda N.16

A quale tipo di articoli vorresti che C.C.C. dedicasse più pagine?

☐ A Programmi brevi da digitare subito e senza problemi.
☐ B Routine e programmi di utilità che potrebbero essermi utili in futuro.
☐ C Programmi anche lunghi, ma completi e utili (giochi, grafica, musica, utility).
☐ D Articoli di recensioni, prove, novità.

Domanda N.17

Quale tipo di articoli elimineresti volentieri da CCC?

☐ A I programmi molto brevi ma dalle applicazioni troppo limitate.
☐ B I programmi lunghi da digitare.
☐ C Programmi di tipo didattico ma dalle limitate applicazioni pratiche (studi su particolari locazioni, esame di puntatori eccetera).
☐ D Articoli sui computer che attualmente NON possiedi (ricavabili dalle domande 1/X/Y).

Domanda N.18

In quale caso saresti disposto ad accettare l'aumento del prezzo di copertina a L.5000?

☐ A In nessun caso.
☐ B Solo in caso di aumento del numero di pagine.
☐ C Solo in caso di aumento del numero di programmi.
☐ D Solo in caso di incremento di qualità.

Domanda N.19

Quale delle seguenti proposte accetteresti?

☐ A Aumento del prezzo di copertina, ma inserimento del dischetto (1541 compatibile e non protetto) contenente i programmi pubblicati ed altri file (cioè "Directory")
☐ B Aumento del prezzo di copertina, ma inserimento del nastro (C/64 compatibile e non protetto) contenente i programmi pubblicati.
☐ C Lasciare C.C.C. così come è adesso
☐ D Aumentare il numero di pagine dedicate alla didattica in senso generale (Assembly, C, Pascal, Basic, eccetera)

Domanda N.20

Supponendo di entrare in possesso di altri computer, che rivista ti piacerebbe trovare in edicola?

☐ A Una simile a C.C.C. ma specifica per tali computer
☐ B Una rivista che contenga un dischetto
☐ C Una rivista che aiuti a conoscere il computer
☐ D Una rivista che aiuti a utilizzare al meglio il software già disponibile

Domanda N.21

Escludendo l'acquisto di accessori, ricambi e periferiche, quanto spendi ogni mese, mediamente, per "mantenere" il tuo computer? (libri e riviste, nastri e dischi vergini, programmi su supporto magnetico e così via).

☐ A Non più di 10 mila lire al mese.
☐ B Tra le 10 e le 30 mila lire.
☐ C Tra le 30 e le 60 mila lire.
☐ D Oltre le 60000 lire.

Domanda N.22

In che modo, prevalentemente, sei entrato in possesso del software?

☐ A Comprando programmi su supporto magnetico in vendita in edicola.
☐ B Comprando programmi nei negozi.
☐ C Scambiando programmi con altri utenti.
☐ D Uso solo software da me creato o digitato da riviste.

Domanda N.23

Quale è la tua età?

☐ A Minore di 16 anni.
☐ B Compresa tra 16 e 24 anni.
☐ C Compresa tra 24 e 30 anni.
☐ D Superiore ai 30 anni.

Domanda N.24

Quale è la tua occupazione?

☐ A Studente o in attesa di occupazione.
☐ B Tecnico o Impiegato.
☐ C Insegnante o Professionista.
☐ D Artigiano, Operaio o Commerciante.

Domanda n.25

Nel caso tu sia uno studente NON universitario, che corso di studi stai seguendo (oppure: che titolo di studi possiedi)?

☐ A Scuola dell'obbligo (elementari, medie)
☐ B Liceo classico, scientifico
☐ C Istituto tecnico (per geometri, ragionieri, corrispondenti, eccetera)
☐ D Istituto professionale o scuola di tipo diverso dai precedenti

Domanda n.26

Nel caso tu sia uno studente universitario, a quale facoltà risulti iscritto (oppure, se sei laureato: che diploma di laurea hai conseguito)?

☐ A Lettere, Giurisprudenza o altre facoltà umanistiche
☐ B Matematica, Scienze o altre facoltà scientifiche
☐ C Ingegneria o altre facoltà tecniche
☐ D Corso para-universitario o post-diploma

Domanda N.27

Da quanto tempo possiedi e/o usi un computer?

☐ A Da oltre due anni.
☐ B Da oltre un anno.
☐ C Da pochi mesi.
☐ D Da pochissimo tempo (oppure: non lo possiedo).

Domanda N.28

Che tipo di pubblicazioni di informatica preferisci?

☐ A Riviste che insegnino a programmare e a conoscere il computer.
☐ B Riviste su supporto magnetico (nastro o disco).
☐ C Pubblicazioni che informino sulle novità riguardanti qualsiasi computer.
☐ D Pubblicazioni che parlino solo del computer che possiedi.

▽▲▽

DOMANDE EXTRA

Quale è il primo numero di C.C.C. che ricordi di aver letto?

☐ N..............

Quali riviste di informatica acquisti abitualmente (almeno quattro numeri l'anno):

☐ Commodore Computer Club
☐ Personal Computer
☐ Micro & Personal Computer
☐ MC Microcomputer
☐ Bit
☐ Noi 64 & C/128
☐ Commodore Gazette
☐ Computers & Electronics
☐ Sperimentare col computer
☐ Elettronica e computer
☐ List
☐ Chip
☐ Zapp

▽▲▽

Stacca le pagine relative all'inchiesta ed inviale, in busta chiusa affrancata secondo le norme vigenti, indirizzando a:

Systems Editoriale
Viale Famagosta, 75
20142 Milano

Cognome ______________________

Nome ______________________

Indirizzo ______________________

Città ______________________

Cap ______________________

Prefisso/Telefono ______________________

818 ($332) / 1138 ($472):
Sono 321 byte riservati ai buffer di cassetta ed all'interfaccia RS-232; chiaramente sarà possibile l'utilizzo di tutti i 321 byte se non si adopera nè il registratore a cassette, nè la cosiddetta User Port. Se necessitano utilizzazioni parziali di queste periferiche, si sappia che il confine tra le due aree dedicate è rappresentato dalla locazione 1014 ($3F6) che appartiene alla "zona registratore", mentre dalla 1015 ($3F7) in poi è presente il buffer per la porta utente.
1375 ($55F) / 1984 ($7C0):
Ben 610 byte tra area di definizione dei tasti programmabili e zona dedicata alle routine di banking così suddivisi:

da 1375 ($55F) a 1432 ($598) per il C/16 e fino a 1443 ($5A3) nel Plus/4: rispettivamente 58 e 69 byte nei quali vengono definiti i comandi assegnati ai tasti funzione (Help compreso); è possibile immettervi dati solamente se non si prevede l'uso dei tasti suddetti, anche se rimane possibile rilevarne l'eventuale pressione dalle locazioni 198 ($C6) e 2038 ($7F6) secondo i codici riportati nell'apposito riquadro.

Dal termine dell'area precedente a 1515 ($5EB): rispettivamente, 83 (C/16) e 72 (Plus/4) byte con la preziosissima caratteristica dell'immunità al reset.

Da 1516 ($5EC) a 1519 ($5EF): 4 byte usati dalla "tabella degli indirizzi fisici"; sono sovrascrivibili ma vengono modificati da un eventuale reset.

Da 1520 ($5F0) a 1524 ($5F4): 5 byte immuni al reset.

Da 1525 ($5F5) a 1544 ($608): solamente nel Plus/4, quest'area è occupata da una routine di 20 byte che attiva la R.O.M. contenente il programma 3-PLUS-1 incorporato; nel Commodore 16 le stesse locazioni sono libere come la zona successiva.

Da 1545 ($609) a 1791 ($6FF): ben 247 byte consecutivi immuni al reset, posti in un'area riservata, in teoria, al sintetizzatore vocale (!?) e, nella parte finale, all'inizio di uno pseudo-stack per l'interprete Basic implementato, nella versione 3.5, sul C/16 e sul Plus/4.

1792 ($700) / 1984 ($7C0):
Altri 193 byte, sempre consecutivi ai precedenti, con il difetto di essere modificabili dal tastino di reset.

3048 ($BE8) / 3071 ($BFF):
24 byte compresi tra la fine della zona riservata agli attributi di colore dello schermo e la RAM video vera e propria. Sullo schermo vi sono 1000 (40 colonne x 25 linee) diverse cellette, ciascuna con un colore proprio: a tale scopo il computer dedica un intero Kilobyte (1024 byte) da 2048 ($800) a 3071 ($BFF); le nostre 24 locazioni sono, appunto, gli ultimi byte inutilizzati ed hanno anche il vantaggio dell'immunità al reset.

4072 ($FE8) / 4095 ($FFF):
24 byte, analoghi ai precedenti, posizionati tra la fine della RAM video (contenente i codici dei caratteri attualmente visualizzati sullo schermo) e l'inizio standard della memoria riservata al testo Basic fissata a 4097 ($1001), in quanto la 4096 è sempre a zero, pena l'emissione di ?SYNTAX ERROR vari; anche questa zona, come la precedente, gode della caratteristica di non subire alterazioni in seguito ad un reset.

Come si sarà ben compreso, quindi, la caratteristica di immunità al reset è tra le più preziose per la scelta della RAM nella quale sistemare i programmi in L.M. in quanto, essendo tale linguaggio totalmente privo di controlli, in presenza di qualche errore il computer si impianta nella stragrande

A volte è bene evitare l'uso di alcuni comandi Basic per non distruggere le routine l.m. allocate in zone RAM "pericolose"

Una routine l.m. può anche essere "spezzata" in più segmenti collegati tra loro mediante JMP

maggioranza dei casi, richiedendo così l'intervento del magico tastino di reset per sbloccare la situazione.

Tornando a parlare di RAM, se ne trova ancora in un'altra zona, ben lontana da quella Basic: si tratta di 30 byte nell'area riservata al TED Chip da 65312 ($FF20) a 65341 ($FF3D) e di altri 182 byte da 65344 ($FF40) a 65525 ($FFF5) nella memoria ad accesso casuale situata "sotto" la ROM presente agli stessi indirizzi: per l'uso di quest'ultima zona, però, è estremamente importante conoscere il funzionamento del registro numero 62 posto a 65342 ($FF3E) e del "fratello" numero 63 posto a 65343 ($FF3F).

Entrambi, oltre alla locazione CURBNK posta a 251 ($FB) per l'attuale configurazione dei banchi, controllano la configurazione di memoria del sistema e sono del tipo "Write only", cioè vanno adoperati solo per scriverci qualche valore in quanto, agendo come degli interruttori, non ha senso leggerne i contenuti: scrivendo nel registro numero 62 si provvede alla selezione della ROM nel banco di memoria compreso tra 32768 ($8000) e 65535 ($FFFF), escludendo contemporaneamente sia le zone relative a procedure di ingresso / uscita che il TED da 64976 ($FD00) a 65343 ($FF3F).

Scrivendo, invece, nel registro numero 63, si seleziona la RAM (se c'è...) nello stesso intervallo di memoria; pertanto il Basic ed il sistema operativo (cioè tutta la ROM) possono essere disabilitati; vediamo subito un esempio chiarificatore su come si possa (da L.M.) disabilitare il reset su un C/16 & Plus/4.

Dopo aver digitato MONITOR e premuto il tasto di Return si introduca il disassemblato n. 1.

Per chi avesse dimenticato la gestione di "Monitor", ricordiamo che alla pressione del tasto Return il puntino posto sulla sinistra (che compare prima della locazione di memoria) della linea successiva, si tramuterà nella lettera "A": niente paura, indica solamente che si sta Assemblando un programma in L.M.

Terminata l'operazione si può tornare in ambiente Basic (con "X" e Return) per mandare in esecuzione il miniprogramma in assembly appena digitato, con una semplice SYS 8192 che è il valore decimale di $2000 da dove, appunto, inizia il nostro listato. Il tasto di reset non terrà più fede al suo nome mentre lo schermo darà luogo ad uno sfolgorio continuo di colori differenti.

A parte (disassemblato n. 2) riportiamo i commenti al programma precedente grazie ai quali sarà più facile assimilare la corretta modalità di gestione della memoria "alta" posta sotto la ROM per mezzo dei due potenti registri 62 e 63 del Commodore 16 & del Plus/4.

Come si nota, l'uso di questi registri non è poi così complicato; il 63 ($FF3F) richiede tuttavia qualche attenzione in più in quanto, disabilitando totalmente la ROM, il programmatore deve pensare a gestire tutta una serie di particolari ai quali, normalmente, non presta attenzione come, ad esempio, l'interrupt.

Se non si ha necessità di una tale routine, è meglio disabilitare gli IRQ con l'apposita istruzione SEI (SEt Interrupt), altrimenti basta ricordarsi di inserire la routine stessa da qualche parte in memoria ed immettere l'indirizzo di partenza relativo nelle locazioni 65534 ($FFFE) e 65535 ($FFFF) nell'usuale forma byte basso / byte alto: tutto qui.

Per il resto basta un po' di fantasia per creare, addirittura, sistemi operativi personalizzati. Purtroppo non tutti i Commodore 16 hanno 64 Kbyte di memoria, altrimenti sarebbe già stato pubblicato un Basic V 3.5 totalmente in italiano e perfettamente compatibile con quello tradizionale in inglese.

LA RAM LIBERATA

Un accurato elenco delle locazioni libere (o in libertà vigilata) per allocare proprie routine l.m. nella memoria del C/128

di **Alessandro Diano**

Il C/128 mette a disposizione dell'utente quantità incredibili di memoria RAM

Il calcolatore "ibrido" C/128, come ben si sa, è dotato di due microprocessori e di tre sistemi operativi.

Uno dei tre risulta essere quello dell'home computer più venduto al mondo (C/64). Ciò consentiva, al C/128, di disporre di qualche migliaio di programmi prima ancora della sua commercializzazione.

Molti dei discorsi fatti sull'hardware del C/64, infatti, sono spesso validi anche per il C/128, a parte lievi modifiche, solitamente inerenti alcune locazioni in pagina zero.

Ecco, pertanto, che risulta patetico parlare del raster specifico per il C/128 quando esiste fior di documentazione in merito al C/64. Una delle routine classiche è certamente quella che visualizza la pagina grafica in contemporanea alle righe di testo. Il problema, difficile da ottenere in un C/64, risulta essere brutalmente eliminato grazie ad una semplicissima istruzione Basic (GRAPHIC 2, 1, X) con la quale si può persino decidere da quale riga iniziare l'area testo.

Tenendo presente che risulta inutile scrivere routine l.m. già implementate in comode istruzioni Basic, riporteremo, tuttavia, l'indicazione dello spazio RAM opportuno per sistemare proprie creazioni "extra - Basic".

La memoria antecedente l'inizio del testo in linguaggio Basic è molto ampia (c'è spazio sino alla locazione 7167 = $1BFF); per questo motivo sono state prese in considerazione esclusivamente le zone caratterizzate dall'utile proprietà di non essere influenzate da eventuali reset, volontari o accidentali che siano.

RAM PRE-BASIC PER C/128

Ecco, qui di seguito, la descrizione della zona RAM più interessante per approfondire gli studi accennati in precedenza:

269 ($10D) / 283 ($11B):
15 byte nell'area di lavoro del DOS per la conservazione di altrettanti dati oppure, al massimo, per una mezza dozzina di istruzioni di linguaggio macchina.

285 ($11D) / 448 ($1C0):
164 byte nello stack dell'8502; è opportuno non esagerare nello sfruttare questa zona sino in fondo, onde evitare di trovare i propri dati alterati da una miriade di GOSUB, FOR / NEXT, JSR e "push" vari.

528 ($210) / 672 ($2A0):
145 byte nel buffer di ingresso comandi per il monitor ed il Basic; se il programma sta girando è un'ottima area per programmi "volatili", in caso contrario è meglio evitarne l'uso.

888 ($378) / 895 ($37F):
8 byte nella tabella degli indirizzi secondari;

I CODICI DI TASTIERA SUL C/128

Riportiamo i codici di tastiera del C/128 i quali possono essere letti dalle locazioni SFDX ed LTSX presenti, rispettivamente, a·212 ($D4) e 213 ($D5).

Nel Commodore 128 esiste una locazione chiamata SHFLAG che, relativa all'indirizzo 211 ($D3) della pagina zero, presenta i seguenti valori a seconda del tasto premuto:

0 (Nessuno), 1 (Shift), 2 (Commodore), 4 (Control), 8 (Alt), 16 (Ascii / Cc oppure Caps Lock).

Rimane valida la regola secondo cui un valore differente da quelli riportati indica la pressione, in contemporanea, dei tasti i cui codici relativi sono gli addendi del valore letto.

Una routine l.m. può essere allocata in numerose zone di memoria

se non se ne usano troppi vi si possono memorizzare valori di lavoro, tipo "usa e getta".

996 ($3E4) / 1006 ($3EE):
11 byte in un'area libera; è una zona sicura per dati importanti.

2024 ($7E8) / 2039 ($7F7):
16 byte analoghi all'area precedente.

2048 ($800) / 2559 ($9FF):
ben 512 byte di "pseudo-stack" del Basic; è da usare con cautela nel caso di un massiccio impiego dei comandi in V.7.0.

2670 ($A6E) / 2734 ($AAE):
65 byte, non documentati, pronti da riempire!

2743 ($AB7) / 2751 ($ABF):
9 byte identici alla zona precedente.

2758 ($AC6) / 2815 ($AFF):
58 byte posti tra la tabella degli indirizzi fisici ed il buffer di cassetta ad impiego "multiruolo" in quanto possono contenere un miniprogramma L.M. così come messaggi o dati numerici.

3072 ($C00) / 4095 ($FFF):
un intero Kilobyte così suddiviso:

3072 - 3327 ($C00 - $CFF) per il buffer di input della RS-232;

3328 - 3583 ($D00 - $DFF) per il buffer di output della stessa RS - 232;

3584 - 4095 ($E00 - $FFF) per l'area di definizione degli sprite;

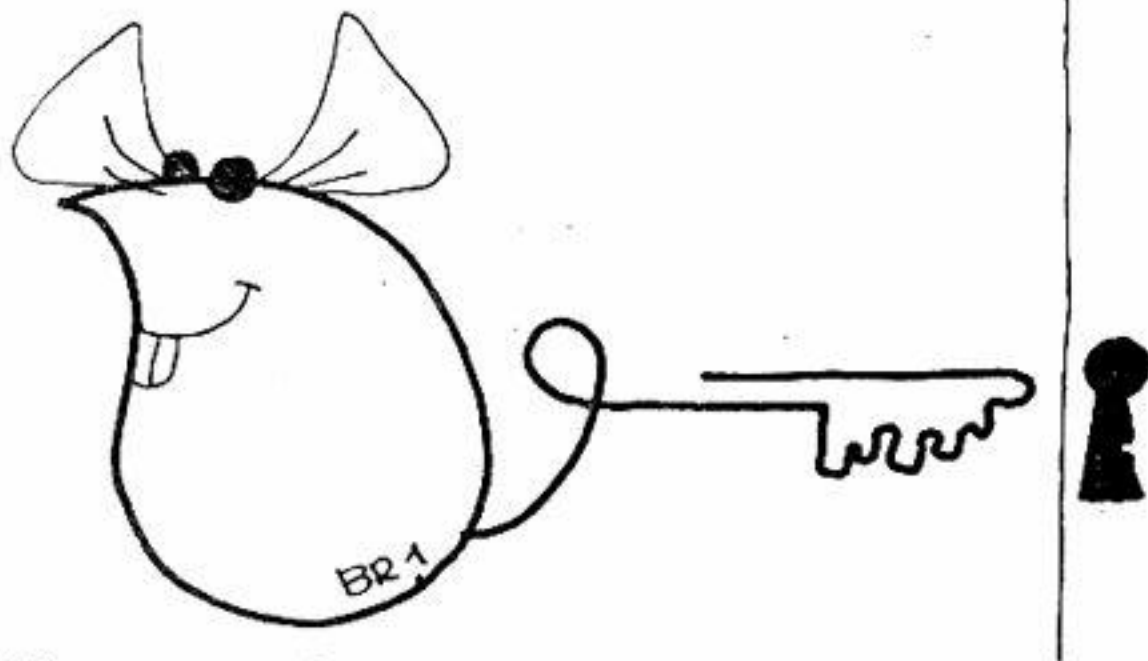
L'utilizzo di RS - 232 e/o sprite sarà direttamente proporzionale al fatto che alcune (o tutte) di tali locazioni si renderanno indisponibili.

4173 ($104D) / 4351 ($10FF):
179 byte nella zona di definizione delle stringhe assegnate ai tasti programmabili; evitando la modifica dei tasti è possibile l'impiego alternativo di tutte le locazioni dedicate tenendo però presente che un reset ripristinerà (sino alla 4172 = $104C) i valori di default.

4360 ($1108) / 4457 ($1169):
98 byte a cavallo tra l'area di messa in opera del comando DOS (sino alla 4400 = $1130) e l'area di lavoro grafica (dalla locazione 4401 alla 4462 = $116E); ci si sappia regolare in caso di utilizzo di tali feature dell'elaboratore.

Come sempre, al termine di ogni missione, Primo si reca nella saletta di proiezione ($28C0) per vedere, insieme agli altri ufficiali, il filmato del suo volo, che era stato ripreso dalla cinepresa di bordo del suo jet.

	VALORE LETTO		43	$2B	-
			44	$2C	.
Decim.	Esad.	TASTO PREMUTO	45	$2D	:
0	$00	Delete	46	$2E	Chiocciolina
1	$01	Return	47	$2F	,
2	$02	CRSR orizzontale	48	$30	£
3	$03	F7 / F8	49	$31	*
4	$04	F1 / F2	50.	$32	;
5	$05	F3 / F4	51	$33	Home
6	$06	F5 / F6	52	$34	CHR$(1)
7	$07	CRSR verticale	53	$35	=
8	$08	3	54	$36	Freccia in alto
9	$09	W	55	$37	/
10	$0A	A	56	$38	1
11	$0B	4	57	$39	Freccia a sinistra
12	$0C	Z	58	$3A	CHR$(4)
13	$0D	S	59	$3B	2
14	$0E	E	60	$3C	Spazio
15	$0F	CHR$(1)	61	$3D	CHR$(2)
16	$10	5	62	$3E	Q
17	$11	R	63	$3F	Run/Stop
18	$12	D	64	$40	Help
19	$13	6	65	$41	8 [Tastierino]
20	$14	C	66	$42	5 [Tastierino]
21	$15	F	67	$43	Tab
22	$16	T	68	$44	2 [Tastierino]
23	$17	X	69	$45	4 [Tastierino]
24	$18	7	70	$46	7 [Tastierino]
25	$19	Y	71	$47	1 [Tastierino]
26	$1A	G	72	$48	Escape
27	$1B	8	73	$49	+ [Tastierino]
28	$1C	B	74	$4A	- [Tastierino]
29	$1D	H	75	$4B	Line Feed
30	$1E	U	76	$4C	Enter
31	$1F	V	77	$4D	6 [Tastierino]
32	$20	9	78	$4E	9 [Tastierino]
33	$21	I	79	$4F	3 [Tastierino]
34	$22	J	80	$50	CHR$(8)
35	$23	0	81	$51	0 [Tastierino]
36	$24	M	82	$52	. [Tastierino]
37	$25	K	83	$53	Cursore su
38	$26	O	84	$54	Cursore giu'
39	$27	N	85	$55	Cursore sinistra
40	$28	+	86	$56	Cursore destra
41	$29	P	87	$57	No Scroll
42	$2A	L	88	$58	Nessun tasto premuto

4464 ($1170) / 4473 ($1179):
10 byterelli occupanti le posizioni dei puntatori del comando Basic di RENUMBER (4464 - 4467 = $1173), dei puntatori di lavoro della Directory (4468 - 4471 = $1174 - $1177) e dei valori indice per la grafica (4472 - 4473 = $1178 - $1197); non servendosi di quanto sopra, sono al-

E' necessario prestare una certa attenzione nella scelta della zona più opportuna in cui allocare le routine l.m.

quanto "tranquille".

4589 ($11ED) / 4610 ($1202):
22 byte di cui il primo (4589) è, potenzialmente, il più pericoloso in quanto contiene l'indirizzo secondario del comando Basic RECORD, nel caso di un suo eventuale uso, mentre i rimanenti (sino a 4607 = $11FF) risultano essere liberi.

4659 ($1233) / 4670 ($123E):
12 byte in un'area riservata all'immagazzinamento di puntatori musicali; per il loro uso è da evitare, in generale, l'impiego dei comandi dedicati al suono, con particolare divieto per ENVELOPE.

4701 ($125D) / 4710 ($1266):
10 byte dedicati al modello di ampiezza dell'impulso sonoro; anch'essi sono tabù se necessitano comandi musicali in Basic nei propri lavori di programmazione.
4744 ($1288) / 4860 ($12FC):
117 byte inutilizzabili se si è musicofili informatici; durante l'uso dei vari SOUND e Co., vengono ivi deposte le variabili relative al SID.

4862 ($12FE) / 7167 ($1BFF):
è la zona maggiormente ampia; ben due Kilobyte (ed un quarto) pronti da riempire con programmi in linguaggio macchina anche di discreta lunghezza.

Per quanto riguarda la "tranquillità d'impiego" delle suddette locazioni è opportuno essere a conoscenza delle suddivisioni riportate qui di seguito: i primi due byte (4862 - 4863 = $12FE - $12FF) sono potenzialmente pericolosi nel caso d'uso dei comandi sonori in Basic, in quanto fanno entrambi parte della zona terminale dell'area riservata alle variabili del SID.

L'analisi prosegue da 4864 ($1300) a 6143 ($17FF) con ben 1280 byte assolutamente e splendidamente "scapoli" da ROM di sorta e, quindi, usatissimi per routine quali turbo-tape, estensioni di comandi, protezioni varie etc.; infine, da 6144 ($1800) alla terminale 7167 ($1BFF), è presente una zona di un Kilobyte esatto (1024 byte) ufficialmente dedicato ai tasti funzione anche se, in pratica, è libero per l'utente, analogamente alle 1280 locazioni dell'area precedente.

Sebbene ci si sia riferiti nel corso dell'articolo, in prevalenza, al banco di memoria numero 0, gli "spazi" migliori da adoperare sono quelli precedenti alla locazione 1024 ($400), dal momento che si tratta di una delle due aree in comune ai sedici banchi di memoria del C/128 mentre, come ben sanno gli utenti più affezionati, le altre locazioni, per così dire, "generiche" non godono di cotanto prezioso privilegio.

Si parlava di due zone distinte, ed ecco un netto allontanamento dall'inizio del Basic che, per chi lo sapesse ancora, nel C/128 corrisponde alla locazione 7169 ($1C01).

Terminiamo l'articolo trasferendoci, infatti, in un'area alquanto "inusuale", perchè subito seguente (come posizionamento) ai registri Memory Management Unit (MMU) di configurazione del sistema: 65349 ($FF45) / 65487 ($FFCF).

Sono, questi, gli ultimi 139 byte "liberi" per proprie creazioni Assembly e non: attenzione solamente ad evitare pericolose sovrapposizioni errate che possano mandare in crash il sistema.

"DENTRO" I DRIVE COMMODORE

Anche i 1541 / 71 hanno un'anima da esplorare; le loro ROM, ad esempio...

di **Alessandro Diano**

L'INDISPENSABILE TEORIA

Fin dalla creazione del sistema operativo del Vic 20, la Commodore decise di gestire le periferiche esterne in maniera piuttosto semplice, assegnando ad ogni device un numero che lo caratterizzasse senza equivoci; la tastiera assunse il numero 0, il registratore a cassette il n.1, la RS-232 il 2 e lo schermo il 3.

Per non limitare la versatilità del sistema nei confronti dei futuri dispositivi che fossero stati presentati sul mercato, si decise di dotare il computer di un bus seriale (che l'ennesima commodorata di mamma C.B.M. ha voluto essere rigorosamente NON standard) verso il quale indirizzare tutti i "device number" maggiori di 3.

Il risultato è stato quello delle cosiddette periferiche intelligenti, dotate cioè di un proprio Sistema Operativo (S.O.) che interpreta quanto viene inviato attraverso il bus e di un proprio microprocessore (addirittura due nelle unità a disco doppie) che rende il dispositivo esterno paragonabile al computer che lo utilizza: R.A.M, R.O.M. e dispositivi I/O inclusi.

Indubbiamente i vantaggi di una tale scelta (diametralmente opposta a quella sviluppata fino allora da altre marche) sono molteplici: la periferica è, anzitutto, in grado di riconoscere ed interpretare in maniera autonoma i comandi inviati; se, poi, ne viene realizzata una nuova versione, dalle caratteristiche migliorate o innovative, non sarà necessario cambiare il sistema operativo del computer che utilizza tale periferica, in quanto le procedure per la gestione dei vari comandi saranno contenute nel dispositivo esterno stesso. Ecco perchè, ad esempio, la lettera "V" inviata sul bus seriale indica ad un drive di riorganizzare la mappa della disponibilità dei blocchi in funzione della directory del dischetto (la famosa Validate), mentre viene considerato un carattere come un altro da una stampante o da un plotter: il loro S.O. non contempla l'interpretazione della "V" come comando.

Ed ancora, parlando del solo disk drive, è sempre lo stesso motivo che permette al possessore di un 1571 di cambiare il numero del proprio device con la sequenza...

OPEN [numero file], [numero di device attuale], 15, "U0>" + CHR$ ([nuovo numero di device desiderato]): CLOSE [numero file]

Esplorare le ROM del drive è uno dei sogni ricorrenti degli esperti in Assembly

Nell'articolo è riportata una breve ma succosa mappa della memoria del 1541

...mentre non lo consente all'utente del 1541: nella R.O.M. del drive a doppia faccia è semplicemente presente l'apposita routine che interpreta correttamente il messaggio "U0>" etc. che, come si intuisce, è assente nel S.O. del drive a singola faccia (1541) il quale non intraprende alcuna azione utile in seguito al comando sopra riportato.

Un altro dei vantaggi derivanti dall'impiego delle periferiche intelligenti è dato dalla possibilità di disimpegnare velocemente, dalle operazioni di I/O, l'unità centrale (il computer), inviando i dati tutti in una volta all'interno di un apposito buffer (che la periferica deve ovviamente prevedere e, tendenzialmente, prevede) e lasciando che sia il S.O. del dispositivo esterno a sbrigare il resto del lavoro quale, ad esempio, stampare i dati su carta, registrarli su supporto magnetico e così via.

Tanto per curiosità, ricordiamo che i progettisti del sistema operativo del 1571 sono stati i signori David Siracusa, sicuramente di origine italiana, e Greg Berlin che nel 1985 hanno creato, rispettivamente, il software e l'hardware del drive.

Forse, come elemento negativo di una periferica intelligente, possiamo indicare l'impossibilità di "esplorare" le sue funzioni, operazione altrimenti possibile disassemblando le ROM del computer qualora le contenesse. Volendo, ad esempio, esaminare il comando COLLECT (disponibile sul C/128) si trova, infatti, la sequenza di Open... Print#... Close che lascia a bocca asciutta anche i più fanatici assembly-sti.

QUI COMINCIA L'AVVENTURA

La struttura dei due disk drive Commodore più diffusi, vale a dire il 1541 ed il 1571, è sostanzialmente piuttosto semplice; entrambi sono controllati dal microprocessore 6502-A della MOS Technology e da due chip d'interfaccia I/O e temporizzazione interna denominati V.I.A. (Versatile Interface Adapters) 6522-A; possiedono una R.O.M di 16 Kilobyte per quanto riguarda il 1541 e di 32 Kb. per il 1571 ed hanno entrambi una R.A.M. di 2 Kb. dei quali più della metà (esattamente 1280 byte) è suddivisa in cinque buffer da 256 byte ciascuno a disposizione (volendo) dell'utente.

I due V.I.A. sono presenti, rispettivamente, nelle locazioni dalla 6144 (esadecimale $1800) alla 6159 ($180E) per quanto riguarda il chip del bus seriale (V.I.A. numero 1), mentre dalla 7168 ($1C0Q) alla 7182 ($1C0E) si trovano le locazioni del chip che controlla il motore del drive e la testina di lettura / scrittura (V.I.A. numero 2).

Entrambe le R.O.M. terminano, ovviamente, alla locazione 65535 ($FFFF) ed iniziano, per il 1541 ed il 1571 rispettivamente, la prima dalla 49408 ($C100) mentre la seconda ben 16 Kb. prima, cioè dalla 32768 ($8000).

In quest'ultimo S.O. sono in realtà presenti due ROM per mantenere la compatibilità tra i drive: infatti da 32768 a 49407 ($C0FF) è presente la nuova R.O.M. propria del 1571, mentre da 49408 in poi è stato ricopiato pari pari (salvo alcuni aggiornamenti dei quali parleremo la prossima volta) il sistema operativo del 1541; in tal modo è così possibile far girare anche i programmi che impiegano i comandi Memory riferiti al drive a singola faccia.

Tornando sul discorso R.A.M. i cinque buffer, numerati da zero a quattro, sono allocati agli indirizzi riportati in tabella 1.

BUFFER	INDIRIZZO DEC.	INDIRIZZO ESA.
0	768-1023	$300-$3FF
1	1024-1279	$400-$4FF
2	1280-1535	$500-$5FF
3	1536-1791	$600-$6FF
4	1792-2047	$700-$7FF

Tabella 1

Normalmente il buffer numero 4 non è disponibile in quanto vi è immagazzinata la B.A.M, ovvero la "Block Availability Map" (mappa della disponibilità dei blocchi) di cui il settore 0 della traccia 18 nel 1541 ed i settori 0 delle tracce 18 e 53 nel 1571; se, contemporaneamente, si lavora con file di tipo REL e SEQ (relativi e sequenziali), anche il buffer 3 viene impiegato dal S.O. per la directory.

E' proprio in tali buffer che vengono eseguite quelle meravigliose routine che formattano un dischetto nel tempo che Calvin Smith impiega a fare i cento metri piani, che leggono e scrivono in tracce oltre la 35ma e che, insomma, fanno fare al drive tutte quelle diavolerie normalmente impossibili.

Questa zona R.A.M, inoltre, è stata anche intelligentemente organizzata grazie ai "Comandi Utente" dedicati che vanno da "U3" a "U8", permettendo il posizionamento, dalla locazione 1280 ($500) alla 1295 ($50F), di ben sei JMP (acronimo inglese del termine JuMP cioè salto, equivalente, in assembly, al comando basic GOTO). Ciò permette di accedere a routine sia nostre (R.A.M.) che del S.O. (R.O.M.), creando una specie di Jump Table (tavola dei salti) simile a quella contenuta nei computer dalla locazione 65409 ($FF81) alla 65523 ($FFF3) attraverso la quale si accede alle routine del kernal.

Nel caso dei drive, con un semplice Ux (con x compreso tra 2 e 9) è possibile accedere ad una qualunque routine "puntata" dal JMP interessato dal comando U.

LA R.A.M. QUESTA SCONOSCIUTA

La zona decisamente più interessante, nella quale mettere il naso, è sicuramente la R.A.M. che va dalla locazione 0 ($0) alla 767 ($2FF) in quanto la sua comprensione presuppone la conoscenza delle routine R.O.M. che ne alterano i valori: una specie di invito obbligato all'approfondimento, insomma.

Partendo dall'inizio, le prime cinque locazioni (numerate da 0 a 4) che vanno sotto il nome di JOB QUE, sono di una potenza spaventosa: la routine presente da 65127 ($FE67) in poi, volgarmente conosciuta come Interrupt, le analizza di continuo in attesa di trovarvi uno dei codici di comando riportati in tabella 2. Attenzione! non sono i comandi tipo i ben noti "M-W", ma i veri e propri codici che il disk controller esegue.

In pratica sono come delle istruzioni elementari di linguaggio macchina, a differenza di "M-W" che, nell'analogia, si può paragonare ad un comando basic.

CODICE	OPERAZIONE	DESCRIZIONE
128-$80	Read	Lettura dei dati dal disco
144-$90	Write	Scrittura dei dati sul disco
160-$A0	Verify	Verifica dei dati sul disco
176-$B0	Seek	Ricerca di un blocco particolare
192-$C0	Bump	Allineamento della testina
208-$D0	Jump	Salto ad una routine in L.M.
224-$E0	Execute	Salto con ritorno ad una routine L.M.

Tabella 2

Al termine dell'operazione, la stessa locazione da 0 a 4 (secondo il buffer scelto) conterrà uno dei codici di errore riportatti in tabella 3.

Chiaramente è necessario, per alcuni comandi, il passaggio di ulteriori parametri (oltre al buffer scelto ed al codice di comando) quali, ad esempio, la traccia ed il settore ove effettuare l'operazione richiesta. Le locazioni 6 e 7 forniscono, rispettivamente, proprio queste due indicazioni per il buffer 0; la 8 e la 9 per il buffer 1, e così via sino ad arrivare alla coppia 14 / 15 ($0E / $0F) relativa al quarto ed ultimo buffer.

Attivare una routine l.m. è la cosa più semplice del mondo

CODICE	ERRORE	DESCRIZIONE
1-$01	OK	Nessun errore
2-$02	Read	Intestazione del blocco non trovata
3-$03	Read	Carattere di sincronismo non trovato
4-$04	Read	Blocco dati mancante
5-$05	Read	Checksum errata nel blocco dei dati
6-$06	Read	Decodifica errata
7-$07	Write	Verifica errata
8-$08	Write	Disco protetto in scrittura
9-$09	Read	Checksum d'intestazione errata
10-$0A	Read	Sovrascrittura dei dati sul blocco successivo
11-$0B	Read	Identificatore del disco non corretto

Tabella 3

Altre locazioni degne di considerazione nella R.A.M. dei disk drive sono la 18 / 19 ($12 / $13) e la 22 / 23 ($16 / 17) che contengono i due caratteri dell'identificatore del dischetto attuale (ID) per il drive 0; la 119 ($77) che contiene il numero di periferica sommato alla costante 32 ($20) per i comandi di LISTEN, la 120 ($78) che è simile alla precedente ma ha una costante di 64 ($40) e si riferisce ai comandi di TALK, la 106 ($6A) che contiene il numero di tentativi di lettura in caso di errore (generalmente contiene cinque ma, usando programmi commerciali, è meglio modificarla ad uno per salvaguardare l'allineamento delle testine del proprio drive). Il gruppo di locazioni che va dalla 127 ($7F) alla 132 ($84) contengono, rispettivamente, il numero del drive (zero oppure uno), l'attuale traccia, settore, numero di file ed indirizzo secondario.

Da 512 ($200) a 552 ($228) si trova il buffer di input per la stringa di comando inviata al drive che, essendo lungo 42 byte, limita a tale lunghezza il massimo dei caratteri inviabili per volta (comandi M, B compresi); da 689 ($2B1) a 724 ($2D4) vi è il buffer per la directory e, in particolare, il nome dei file si trova da 691 ($2B3) a 706 ($2C2), in 708 ($2C4) l'eventuale asterisco per i file non chiusi correttamente, da 709

DAL REGISTRATORE AL DRIVE

Una delle principali motivazioni per le quali nel lontano 1985 decisi di abbandonare il datassette in favore di un 1541 (per giunta usato), fu sicuramente la soddisfazione nel notare che, a parte il cambio di unità di misura del tempo di attesa, la gestione I/O del sistema passava ad un livello decisamente superiore.

Ne guadagnai anche in salute, grazie alla diminuita frequenza di comparsa del messaggio LOAD ERROR dopo ANNI (o almeno a me parevano tali) passati davanti ad uno schermo completamente azzurro che non dava segni di vita; inoltre riuscivo a trovare il "mitico" programma di nome PIPPO tra la miriade di suoi colleghi, senza incrementi rilevanti del diametro delle mie coronarie: sicuramente un gran bel vantaggio!

Ben presto, però, ci si rende conto che un drive è qualcosa di più di un semplice registra-programmi: possiede al suo interno un tesoro di routine (ovviamente in linguaggio macchina) che permettono all'utente che non legga i soli manuali Commodore di avere a disposizione una gran quantità di caratteristiche interessanti, tra le quali i file relativi, quelli sequenziali, i comandi per il disco e quelli per l'accesso diretto sono solo i più famosi o, meglio, i meno sconosciuti.

($2C5) a 711 ($2C7) le tre lettere che identificano il tipo del file (PRG, SEQ, USR, REL oppure DEL) ed in 712 ($2C8) l'eventuale simbolo di minore (<) per i file protetti dalla cancellazione.

Un altro dei buffer interessanti è quello da 725 ($2D5) a 761 ($2F9) che contiene i messaggi d'errore che verranno poi inviati al computer; così come le locazioni 762 ($2FA) e 764 ($2FC) contenenti il byte basso e quello alto dei blocchi presenti sul dischetto: è chiaro, però, che tutte quelle locazioni che danno informazioni sul dischetto vanno lette, come minimo, dopo l'invio di un Initialize.

Si hanno ora gli elementi di base per applicare il tutto: con l'aiuto di un monitor per drive (ne dobbiamo pubblicare uno oppure ne siete già in possesso?) si introduca il seguente disassemblato L.M. oppure, in mancanza del monitor, l'equivalente listato basic, entrambi riportati in queste pagine.

```
ASSEMBLY:

0300 LDA #$12  Traccia 18
0302 STA $08   Traccia per buffer 1
0304 LDA #$00  Settore 0 (B.A.M.)
0306 STA $09   Settore per buffer 1
0308 LDA $7F   Numero drive (0 od 1)
030A ORA #$80  Codice di comando Read
030C STA $01   JOB QUE per buffer 1
030E JSR $C100 Accende il led rosso
0311 LDA $01   JOB QUE per buffer 1
0313 BMI $0311 Attende codice d'errore
0315 CMP #$02  Errore minore di due?
0317 BCC $031E Si, esci
0319 LDX $7F   Numero drive
031B JMP $E60A Gestione dell'errore
031E RTS       Esci

Disassemblato n.1
```

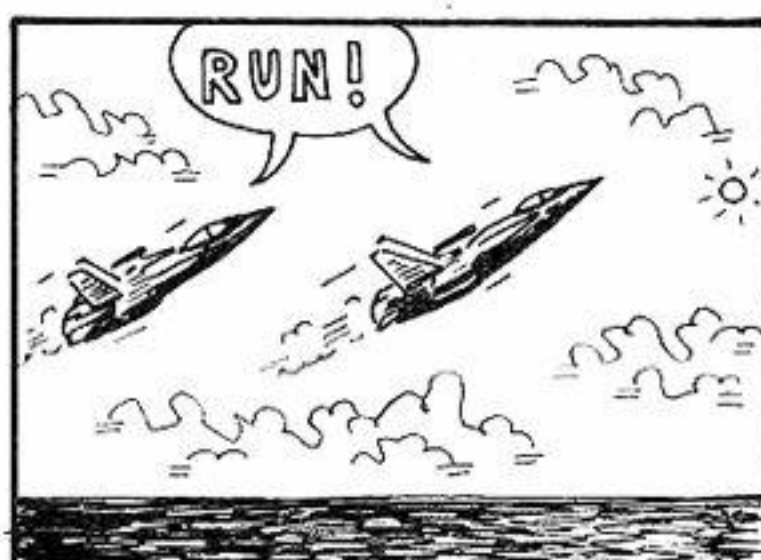
```
BASIC:

 10 DATA 169, 18, 133, 8, 169, 0, 133, 9, 165
 20 DATA 127, 9, 128, 133, 1, 32, 0, 193, 165
 30 DATA 1, 48, 252, 201, 2, 144, 5, 166, 127
 40 DATA 76, 10, 230, 96, -1
 50 OPEN 1, 8, 15, "I0"
 60 READ A: IF A=-1 THEN 90
 70 PRINT# 1,"M-W"; CHR$(B); CHR$(3); CHR$(1); CHR$(A);
 80 B=B+1: GOTO 60
 90 PRINT# 1, "M-E"; CHR$(0); CHR$(3)
100 INPUT# 1, A, B$, C, D
110 PRINT CHR$(147); A; B$; C; D
120 CLOSE 1: END
```

Il programmino di esempio trasferisce nelle locazioni 8 e 9 i valori di 18 ($12) e 0, rispettivamente traccia e settore della mappa della disponibilità dei blocchi; fa quindi un OR tra il numero del drive contenuto in 127 ($7F che, generalmente, contiene zero) ed il comando richiesto (128 = $80 nell'esempio, corrispondente al Read), ponendo infine il risultato nel registro di comunicazione del JOB QUE relativo al buffer 1, la locazione 1, appunto.

Quindi esegue la subroutine da $C100 che accende il led rosso (uno "sfizio" non strettamente necessario ai fini dell'esecuzione del resto del programmino stesso) ed attende che la routine d'interrupt esegua quanto richiesto e ponga il conseguente codice d'errore nel registro di comunicazione del buffer interessato: sempre la locazione numero uno nell'esempio riportato.

Confronta tale codice con il numero due ed esce se è minore (cioè se non vi sono errori), mentre carica il registro X con il numero del drive e salta alla routine di preparazione del relativo messaggio d'errore in caso contrario: al termine dell'esecuzione (pressochè istantanea), nel buffer numero 1, cioè nelle locazioni dalla 1024 ($400) alla 1280 ($4FF) della R.A.M. del drive, sarà presente la Block Availability Map (cioè il settore zero della traccia diciotto) del dischetto inserito nell'unità a dischi.

Per "indurre" il drive in errore potete divertirvi ad inserire, nelle locazioni preposte a contenere i numeri di traccia e settore, valori illegali.

Modificando il breve listato Basic riportato in queste pagine prenderete maggior confidenza con il vostro drive

COMANDI NUOVI: GIOIE E DOLORI

La procedura per aggiungere nuovi comandi Basic è più semplice a farsi che a dirsi

di **Giancarlo Mariani**

Una qualsiasi routine l.m. può esser "convertita in comando Basic

Dopo la pubblicazione dell'articolo "Lo strizzacervelli", che spiegava un metodo per aggiungere nuovi comandi Basic (corredato da un programma che semplficava notevolmente il lavoro), sono giunte in redazione lettere di alcuni lettori che, per inesperienza, non hanno ben compreso i concetti esposti.

Per accontentarli, ecco un "ampliamento" del sopracitato articolo, che spiega per filo e per segno tutte le operazioni da fare, per aggiungere un nuovo comando, tramite il programma pubblicato in precedenza.

Innanzitutto è bene sottolineare che questo articolo può essere compreso da tutti i 64-isti che abbiano seguito, almeno in parte, i vari articoli sul linguaggio macchina; tuttavia non è necessaria una conoscenza approfondita del l.m. del C/64, dal momento che i programmini esposti sono semplicissimi e le spiegazioni ne facilitano la comprensione.

Prenderemo, come esempio, due routine in l.m. semplicissime (ed anche classiche), ossia il cambio del colore dello sfondo ed il cambio del colore del bordo. Ricordiamo che il Basic 2.0 non possiede istruzioni specifiche per ottenere simili scopi, ma bisogna ricorrere alle POKE.

Come sicuramente saprete (è una delle prime cose che si imparano), per cambiare i colori bordo / sfondo occorre alterare il contenuto di due locazioni di memoria, e precisamente la 53280 (bordo) e 53281 (sfondo). Immettendovi un valore compreso tra 0 e 15 (tramite POKE 53280, X e POKE 53281, Y), si otterrà la variazione dei colori dello schermo del 64, secondo la nota tabella presente nel manuale del computer.

Dovendo costruire routine in l.m. che cambiano i colori, queste non dovranno far altro che leggere il parametro che contiene il numero del colore e POKarlo in 53280 (oppure 53281).

Per fare ciò sono sufficienti due routine del sistema operativo:

- $AEFD: Controlla che vi sia una virgola dopo il comando dato: in sua assenza visualizza Syntax error.
- $B79E: Preleva un numero compreso tra 0 e 255 e lo trasferisce nel registro X. Se il numero non è compreso tra 0 e 255 visualizza Illegal Quentity Error.

Costruiamo, quindi, la routine per cambiare il colore del bordo:

JSR$AEFD;Controllo virgola.

```
JSR$B79E;Preleva il numero.
STX$D020;(POKE 53280, X).
RTS;(Torna al Basic).
```

Il primo JSR controlla la virgola, il secondo preleva il numero presente dopo di essa, e STX D020 lo inserisce nella locazione del colore del bordo (D020 esadecimale corrisponde a 53280 decimale).

Allo stesso modo dovrà essere costruita la routine per cambiare il colore dello sfondo, solo che al posto di D020 dovrà essere messo D021 (53281 decimale).

Per caricare in memoria le routine, dovremo ovviamente servirci di un monitor in l.m. o di un assembler (a meno di non caricare i codici decimali delle istruzioni tramite un programmino in Basic). Nel nostro esempio, comunque, ci riferiremo alla prima delle tre soluzioni, ossia al monitor di l.m. e precisamente al ben noto "Zoom", uno dei più diffusi Monitor tra i possessori del 64.

TUTTE LE FASI, MINUTO PER MINUTO

1- Caricare in memoria lo Zoom tramite LOAD "ZOOM", 8, 1 e SYS xxxx. Tale valore (xxxx) è l'indirizzo da cui è rilocato lo Zoom, nel nostro esempio supponiamo che sia 36864, quindi SYS 36864.

2- Digitare le due routine proposte, la prima a partire da $8000 (32768) e la seconda da $C000 (49152). Fate attenzione, comunque, al dislocamento delle sue routine, in modo che non vadano a sovrapporsi alla pagina 0, allo Zoom, o ad altre cose del genere!

Dopo aver digitato le routine, controllate i disassemblati, che dovrebbero, se non avete fatto errori, comparire in questo modo:

```
C000 JSR $AEFD
C003 JSR $B79E
C006 STX $D020
C009 RTS
```

...per la prima, e...

```
8000 JSR $AEFD
8003 JSR $B79E
8006 STX $D021
8009 RTS
```

...per la seconda.

3- A questo punto, verificato che tutto sia a posto, salvate su disco tramite i comandi appositi:

```
S "R1", 08, C000, C00A
S "R2", 08, 8000, 800A
```

Si noti che, nell'esempio che seguiamo, la scelta di due aree di memoria, dotate di indirizzi nettamente differenti, è intenzionale. In questo modo, infatti, imparerete ad inserire nuovi comandi nei casi più disparati possibile. Ci riferiamo sia all'inserimento di routine poste in area "protetta" (come quella a partire da C000), sia all'allocazizone di routine, poste in zona pericolosa, grazie all'alterazione dei puntatori di fine memoria Basic; questa procedura, come è noto, "protegge" da sovrascritture la routine posta a $8000. (Per ulteriori informazioni esaminare i fascicoli precedenti di CCC).

4- Per sicurezza, ora, fate un Reset della macchina o, meglio, spegnetela e riaccendetela.

5- Caricate le due routine salvate prima, tramite:

```
LOAD "R1", 8, 1
```

Il programma cui si fa riferimento è stato pubblicato, in precedenza, su C.C.C.

Non dimenticate di aggiungere il carattere di freccia a sinistra prima di ogni nuovo comando

NEW
LOAD "R2", 8, 1
NEW

Ricordiamo che la seconda routine è stata salvata a partire da $8000; sarà quindi necessario alterare i puntatori di fine memoria Basic, in modo da proteggere la routine dalla sovrascrittura di eventuali stringhe. I puntatori suddetti sono 55 e 56, e, visto che la routine R2 inizia da 32768 ($8000), dovremo settare la fine della memoria a 32767. Quest'ultimo valore, scomposto in byte basso - byte alto, diventa 255 e 127; pertanto...

poke 55, 255: poke 56, 127: new

6- A questo punto provate le due routine impartendo...

SYS49152, b

...per alterare il colore del bordo, e...

SYS32768, s

...per lo sfondo. Naturalmente "b" e "s" sono numeri compresi tra 0 e 15, secondo la solita tabella dei colori.
7- Ora comincia la parte più interessante del lavoro: carichiamo il programma "Command" apparso nel precedente articolo (si suppone che abbiate digitato senza errori e salvato il programma su cassetta o su disco). Una volta caricato, verrà fatto partire tramite RUN.

8- Sul video appaiono alcune opzioni ed alcuni indirizzi: dovremo scegliere l'opzione 3 (Modifica Tabella).

9- Alla richiesta "Dal Numero?", risponderemo 1, mentre "Al numero?", risponderemo 2. Questo significa che vogliamo immettere due comandi nella tabella, appunto il numero 1 ed il numero 2.

10- Ora il programma chiederà, per ognuno dei due comandi, sia il nome che l'indirizzo di partenza: assegnando, ad esempio, al primo comando il nome COLBO (COLore BOrdo) ed al secondo il nome COLSF (COLore SFondo), dovremo inserire:

COLBO [R]

49152 [R]

COLSF [R]

32768 [R]

[R], come al solito, significa che dobbiamo premere il tasto "Return" dopo ogni immissione.
Tornati al menu principale, converrà controllare che tutto sia in ordine: scegliendo un'altra volta l'opzione 3, il programma esegue un listato dei comandi nuovi immessi, ed il cursore si posiziona automaticamente sul primo, permettendo di modificarlo. Se abbiamo commesso un errore, potremo correggerlo come descritto al punto 10, mentre, se è tutto a posto, basterà digitare, come primo carattere del nome del comando, un asterisco (*) per tornare al menu principale.

11- Questa fase è opzionale, ossia potete scegliere se eseguirla o meno: con l'opzione 2 (Salva tabella), il programma registra i comandi inseriti sotto forma di file ASCII, in modo da poterli recuperare, in seguito, così come sono stati scritti, per poterli modificare. Per il momento, dato che i comandi che abbiamo inserito sono di prova e che probabilmente in futuro non ci serviranno, non è necessario salvare la tabella.

12- Anche questo punto, come il precedente, è opzionale, ossia non è necessario eseguirlo per raggiungere lo scopo. Tramite l'opzione 4 (Modifica indirizzi) è possibile modificare l'indirizzo di partenza della routine che interpreta i comandi e della tabella dei nuovi comandi. L'opzione, comunque, si rende necessaria quando gli indirizzi assunti come default (832 per l'interprete dei comandi e 52992 per la tabella) entrano in conflitto con qualche altra routine presente in memoria. Nel nostro caso, quindi, potremo lasciarli così come sono.

13- Questa, invece, è una fase strettamente necessaria: l'opzione 5 salva il l.m. Per suo tramite, infatti, il programma "forma" la routine che dovrà interpretare i comandi, crea la tabella dei nuovi comandi, ed infine salva il tutto su disco.

Come prima cosa si avrà la richiesta "Dal numero... al numero...", alla quale dovremo, ovviamente, rispondere con 1 e 2; in seguito, passato un tempo più o meno lungo nel quale il programma costruisce tutto il l.m. necessario, compare la richiesta: "Nome:". Dovremo rispondere con il nome da assegnare, su disco, alle routine ed alla tabella. Nel nostro esempio, scegliamo il nome LM1.

NB: il nome non può essere più lungo di nove caratteri, poichè il programma aggiunge, in coda al nome, l'estensione, ".R" (per la routine di interpretazione dei comandi), e ".T" (per la tabella), oltre all'indirizzo di partenza.

Al termine viene visualizzato lo status del disco, sia per la routine che per la tabella e, se tutto è andato a buon fine, anche il messaggio:

```
R: OOOKOOOO
T: OOOKOOOO
```

Altrimenti compare l'errore del disco secondo il formato ben conosciuto (numero errore - descrizione - traccia - settore).

14- Ora che il lavoro è praticamente finito, scegliamo l'opzione 6 (Fine), che consente il ritorno al basic e, subito dopo, spegniamo e riaccendiamo il computer.

15- Inseriamo ora il disco in cui avevamo salvato le routine l.m. (da Zoom) e quelle formate dal programma Command. Nella directory dovrebbero comparire i programmi:

```
R1
R2
LM1.R832
LM1.T52992
```

R1 e R2 sono, come ricordiamo, le due routine l.m. caricate da Zoom, mentre LM1.R832 e LM1.T52992 sono, rispettivamente, la routine l.m. di interpretazione dei nuovi comandi, rilocata da 832, e la tabella dei nuovi comandi, rilocata da 52992.

16- Carichiamo... tutto, ossia:

```
load "r1", 8, 1
new
load "r2", 8, 1
new
```

Ricordiamo che dopo aver caricato R2 bisogna, come spiegato prima, spostare i puntatori di fine memoria, cioè:

```
poke 55, 255: poke 56, 127: new
load "lm1.r832", 8, 1
new
load "lm1.t52992", 8, 1
```

Ora rimettiamo a posto i vari puntatori del

Seguendo le varie fasi descritte è possibile creare un set personalizzato di nuovi comandi Basic

Per applicare la procedura descritta è necessara la conoscenza, anche se superficiale, del l.m.

basic, alterati da LOAD, con un bel NEW, impartiamo SYS832, che abilita la routine interprete, e... siamo a posto: possiamo utilizzare i nuovi comandi!

17- Prima dei nomi assegnati ai comandi, ossia COLBO e COLSF, aggiungiamo il simbolo di freccia a sinistra; ricordiamo che la routine di interpretazione dei comandi nuovi esige la freccetta all'inizio di ognuno di essi; siamo pronti per provarli, secondo la sintassi:

```
COLBO, b
COLSF, s
```

IN SEGUITO

Ogni volta che vorremo riutilizzarli non dovremo far altro che ripetere i passi 16 e 17, oppure, ma questo richiede una certa esperienza nella programmazione LM, "unire" routine LM, tabella e routine di interpretazione comandi, in un'unica routine, in modo da non esser costretti a caricare ogni volta i quattro programmi.

Seguendo i 17 passi descritti, si può trasformare, in comando basic, una qualsiasi routine LM, dato che questa, per funzionare come comando, non necessita di alcuna modifica rispetto a quando "lavorava" come SYS.

Un appunto da fare, comunque, riguarda le virgole presenti dopo i comandi nuovi (COLBO, COLSF,). Queste, usate tramite SYS, erano necessarie per separare il comando basic dal parametro; una volta che si considerano tali routine come comandi nuovi, si possono eliminare, rendendo la sintassi dei comandi più elegante, ossia:

```
COLBO b
COLSF s
```

ciò si può ottenere mettendo, al posto del JSR$AEFD (posto all'inizio delle routine), tre NOP (= NO OPeration), ossia il computer, incontrando tale istruzione... non fa assolutamente niente, e passa alla successiva. La routine, così trasformata, diventa:

```
C000  NOP
C001  NOP
C002  NOP
C003  JSR $B79E
C006  STX $D020
C009  RTS
```

Per eliminare la virgola dalle altre routine l.m. (come, ad esempio, quelle dell'enciclopedia di routine LM), basterà, allo stesso modo, mettere al posto del PRIMO (badate bene, SOLO del primo) JSR$AEFD i tre codici NOP.

La modifica, tuttavia, non è essenziale per il funzionamento della procedura.

CALCOLARE MEGLIO DI UN COMPUTER

Come sviluppare, con un qualsiasi computer, calcoli che porterebbero a risultati di 254 cifre!

di **Flavio Molinari**

La "precisione" di un calcolatore è il massimo numero di cifre significative elaborabili nelle operazioni; per ciò che riguarda la parte intera, questa coincide con il limite oltre il quale il numero deve essere rappresentato con la notazione esponenziale, e, per la parte decimale, le cifre dopo la virgola calcolabili senza ricorrere ad arrotondamenti.

Il Commodore 64, analogamente a molti altri computer della sua categoria, può rappresentare sino ad un massimo di 9 cifre decimali e 9 intere, quasi sempre più che sufficienti per normali applicazioni di un home computer, a meno che non si vogliano eseguire calcoli "astronomici", come, ad esempio, quelli richiesti nelle applicazioni di analisi numerica.

Se, come immagino, chi sta leggendo in questo momento non ha (e non avrà mai) problemi del genere, è pregato di considerare la routine proposta come una semplice curiosità, anche se non è detto non debba tornar utile in qualche occasione.

Le operazioni comprese nel sottoprogramma sono moltiplicazioni e potenze sino ad un massimo di 255 cifre significative. I numeri vengono inviati sotto forma di variabili stringa (X1$ e X2$); l'operazione da eseguire, invece, verrà associata alla variabile X0$ in accordo un'intuitiva simbologia: X0$="*" per la moltiplicazione; in tal caso X1$ e X2$ rappresentano i due fattori (solo numeri interi).

Se, invece, ad X0$ si associa il carattere di freccia in alto (tasto compreso tra Restore ed asterisco) si otterrà l'elevamento a potenza. X1$, naturalmente, rappresenterà il valore e X2$ il suo esponente.

Il risultato verrà sempre riportato nella variabile stringa X3$.

Il programma proposto svolge le moltiplicazioni in maniera molto simile al metodo manuale appreso sui banchi di scuola, oggi sempre meno usato a causa (o per merito?) delle minicalcolatrici e computer vari: si prendono le cifre ad una ad una e si moltiplicano fra loro annotando il riporto. Analogamente il programma calcola i prodotti parziali e li memorizza nella matrice XA(), con la differenza che le cifre vengono prelevate a gruppi di quattro per rendere l'algoritmo più veloce.

La prima riga del sottoprogramma consente di usare l'istruzione DIM dimensionando la matrice XA() al primo GOSUB e saltando le istruzioni dopo THEN in tutte le chiamate successive della subroutine.

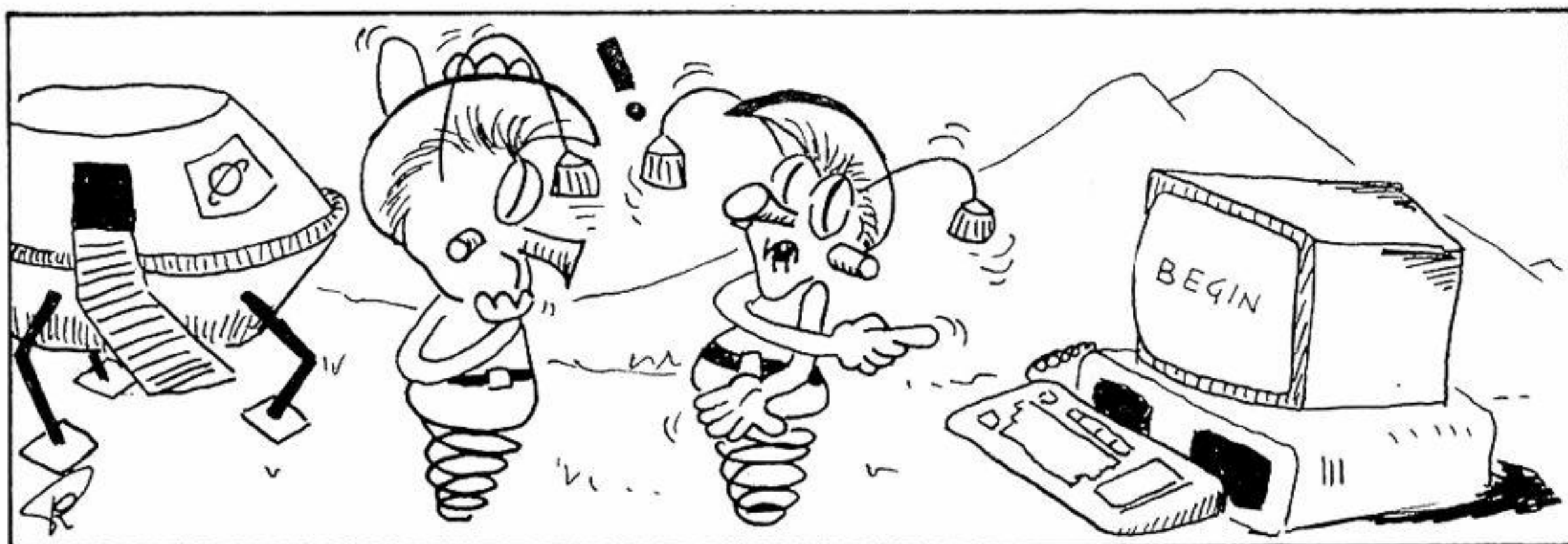
SCHEDA TECNICA

Software per applicazioni matematiche

Hardware richiesto: C/64, C/16, Plus/4, C/128, Vic-20

Consigliato agli esperti

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

```
100 REM MOLTIPLICAZIONI E POTEN
    ZE
110 REM IN MULTIPLA PRECISIONE
130 REM ALGORITMO IN BASIC
140 REM PER QUALSIASI COMPUTER
    COMMODORE
150 REM PARTE DIMOSTRATIVA.
160 DIM XA(65)
170 PRINTCHR$(147)"ECCO LA MOLT
    IPLICAZIONE TRA IL NUMERO..
    ."
180 X1$="12345678901234567890":
     PRINT X1$
190 PRINT:PRINT"...ED IL NUMERO
    ..." :PRINT
200 X2$="67867867867867867867867"
    : PRINT X2$
210 X0$="*" :TI$="000000":GOSUB
     530
220 PRINT:PRINT:PRINT"IL RISULT
    ATO E':":PRINT: PRINT X3$
230 PRINT"(TEMPO IMPIEGATO:"TI$
    " SECONDI)": PRINT
240 FOR I=1 TO 39:PRINT"-";:NEX
    T:PRINT
250 PRINT"ECCO IL NUMERO ";
260 X1$="111": PRINT X1$;
270 PRINT" ELEVATO A ";
280 X2$="15": PRINT X2$
290 X0$="↑" : TI$="000000": GOS
    UB 530
300 PRINT:PRINT"IL RISULTATO E'
    :":PRINT: PRINT X3$:PRINT
310 PRINT"(TEMPO IMPIEGATO:"TI$
    " SECONDI)":PRINT
320 GOSUB 700
330 REM MENU DI SCELTA
340 PRINTCHR$(147)"SCEGLI"
350 PRINT:PRINT"1- MOLTIPLICAZI
    ONE"
360 PRINT"2- ELEVAMENTO A POTEN
    ZA"
370 GOSUB 710
380 IF A$="1" THEN 420
390 IF A$="2" THEN 480
400 GOTO 370
410 REM MOLTIPLICAZIONE
420 X1$="":INPUT"PRIMO FATTORE"
    ;X1$
430 X2$="":INPUT"SECONDO FATTOR
    E";X2$
440 X0$="*":GOSUB530
450 PRINT"RISULTATO:"X3$:GOSUB
    700: GOTO 340
460 REM ELEVAMENTO A POTENZA
470 PRINT:PRINT
480 X1$="":INPUT"NUMERO";X1$
490 X2$="":INPUT"ESPONENTE";X2$
500 X0$="↑":GOSUB530
510 PRINT"RISULTATO:";X3$:GOSUB
     700: GOTO 340
520 REM INIZIO ROUTINE
530 XP=0
540 IF X0$="↑" THEN XP=VAL(X2$)
    :X2$=X1$
550 FOR XT=1 TO XP-1
560 X1=LEN(X1$): X2=LEN(X2$):X3
    $="": FOR Y3=1 TO (X1+X2)/4
    +1: XA(Y3)=0:NEXT
570 FOR Y1=1 TOX2/4+1:FORY2=1TO
    X1/4+1:YF=X1-4*Y2+1:YG=X2-4
    *Y1+1:YZ=YF+3:YV=YG+3
580 IFYF>0THENY8=VAL(MID$(X1$,Y
    F,4))
590 IFYF<=0THENY8=VAL(MID$(X1$,
    1,YZ))
600 IFYG>0THENY9=VAL(MID$(X2$,Y
    G,4))
610 IFYG<=0THENY9=VAL(MID$(X2$,
    1,YV))
620 Y7=Y9*Y8+XA(Y1+Y2-1):XA(Y1+
    Y2-1)=Y7-INT(Y7/10000)*1000
    0
630 XA(Y1+Y2)=INT(Y7/10000)+XA(
    Y1+Y2):NEXT:NEXT:FORY3=(X1+
    X2)/4+1TO1STEP-1
640 XA$=STR$(XA(Y3)):XA$=RIGHT$
    (XA$,LEN(XA$)-1):XA$=RIGHT$
    ("0000"+XA$,4)
650 X3$=X3$+XA$:NEXT
660 IFMID$(X3$,1,1)="0"THENX3$=
    MID$(X3$,2,LEN(X3$)):GOTO66
    0
670 X2$=X3$:NEXT:RETURN
680 REM X1$,X2$,X3$:FATTORI E R
    ISULT.
690 REM MULTIPLA PRECISIONE
700 PRINT CHR$(18)"PREMI UN TAS
    TO PER CONTINUARE"
710 GET A$: IF A$="" THEN 710
720 RETURN
```

UN C/64 TARGATO PASCAL

Anche per il piccolo computer è disponibile una (vecchia) versione Pascal; vediamone, in dettaglio, pregi e difetti

di **Claudio Baiocchi**

Chiunque abbia provato almeno una volta ad accelerare l'esecuzione di un programma Basic, sostituendone la parte più lenta con una routine scritta in Linguaggio Macchina (l.m.) sa benissimo che, da un lato, il Basic Commodore è molto lento: ma che, d'altro lato, il l.m. è altrettanto difficile da manipolare.

Come conseguenza di tutto ciò sembra sensato cercare soluzioni intermedie, provando a lavorare con linguaggi più veloci del Basic, ma più maneggevoli del l.m.

Una delle possibili soluzioni è fornita dal linguaggio Pascal, nel quale la velocità di esecuzione è dovuta al fatto che si tratta di un linguaggio compilato anzichè interpretato; quanto alla facilità di programmazione, la valutazione rimane, ovviamente, soggettiva. Questo articolo non vuole essere un'introduzione al linguaggio Pascal (a tale argomento CCC dedicherà, come promesso in precedenza, una serie di articoli) ma si limita ad illustrare l'implementazione "Oxford Pascal" specifica per il C/64. La versione in nostro possesso è la V 1.0, ma le differenze con altre versioni non dovrebbero essere molte.

L'EDIT DELL'OXFORD PASCAL

Molti lettori posseggono, forse senza saperlo, il linguaggio di cui ci stiamo occupando. Per facilitare la loro ricerca tra i file collezionati(!) riportiamo, qui di seguito, la directory dei file più importanti di Oxford Pascal:

14	"oxford pascal"	prg
101	"pascal lib 01"	seq
61	"pascal lib 02"	seq
121	"pascal lib 03"	seq
14	"pascal lib 04"	seq
121	"pascal lib 05"	seq
18	"pascal error msg"	seq

Per entrare in "ambiente" Pascal occorre, anzitutto, caricare dal dischetto il programma "Oxford Pascal" (di 14 blocchi) e dare il RUN. Schermo e bordo diventano neri; una scritta in verde presenta un messaggio di copyright ed avverte che sta caricando altri file da disco; al termine dell'operazione appare, stavolta in grigio, la familiare scritta "ready.", in minuscolo anzichè in maiuscolo. In questa fase le operazioni sono gestite da un editor un po' diverso da quello standard del Basic; in particolare va notato che:

- alcuni comandi vanno digitati per esteso; ad esempio scrivendo "L" seguito da Shift + I (come siamo abituati in Basic) si genera un Syntax Error (occorre scrivere List per esteso), mentre il punto di domanda (?) funziona normalmente come abbreviazione di Print;
- il comando Let è obbligatorio, cioè la scrittura "X=3" genera Syntax Error, mentre "Let X=3" è accettato; volendo fare a meno del Let occorre premettere il carattere doppio punto (:); con tale artificio altre abbreviazioni vengono accettate.
- certe funzioni non sono più disponibili; ad esempio, il tentativo di usare la funzione Rnd genera un blocco totale del sistema!

COMPILATORI ED INTERPRETI

La distinzione tra "linguaggio interpretato" e "linguaggio compilato" è in realtà impropria: esistono delle versioni del Pascal che funzionano in modo interpretato (una di tali versioni gira su MACINTOSH) così come esistono compilatori per programmi scritti in Basic; anzi esistono, almeno per "grossi" computer, versioni compilate del Basic.

Per capire che cosa significa che il Basic Commodore è "interpretato", occorre rendersi conto che anche un semplicissimo comando quale...
X = Y + 1
...richiede una notevole mole di lavoro; e che la maggior parte di questo lavoro potrebbe essere svolta preliminarmente, traducendo il comando stesso in un linguaggio "più vicino" al Computer.

Ad esempio al computer farebbe più comodo conoscere l'indirizzo delle variabili X ed Y anzichè i loro "nomi": per lui sapere che le variabili si chiamano X ed Y è indifferente: gli serve solo sapere dove andarle a cercare; un discorso analogo si può fare per l'operazione di somma: se indicassimo direttamente l'indirizzo della routine che esegue l'operazione, invece di fornirgli il simbolo "+", il computer potrebbe procedere più speditamente. Se poi il comando in questione deve essere ripetuto più volte (ad esempio se è parte di un ciclo FOR...NEXT) la mole di lavoro risparmiato diviene notevole, ed i tempi di esecuzione diminuiscono sensibilmente.

Il lavoro di un "compilatore" consiste proprio in questo: partendo da un "programma sorgente", cioè da una redazione scritta in termini "umani", il compilatore crea un nuovo programma, scritto in termini più vicini al computer, che prende il nome di "programma oggetto".

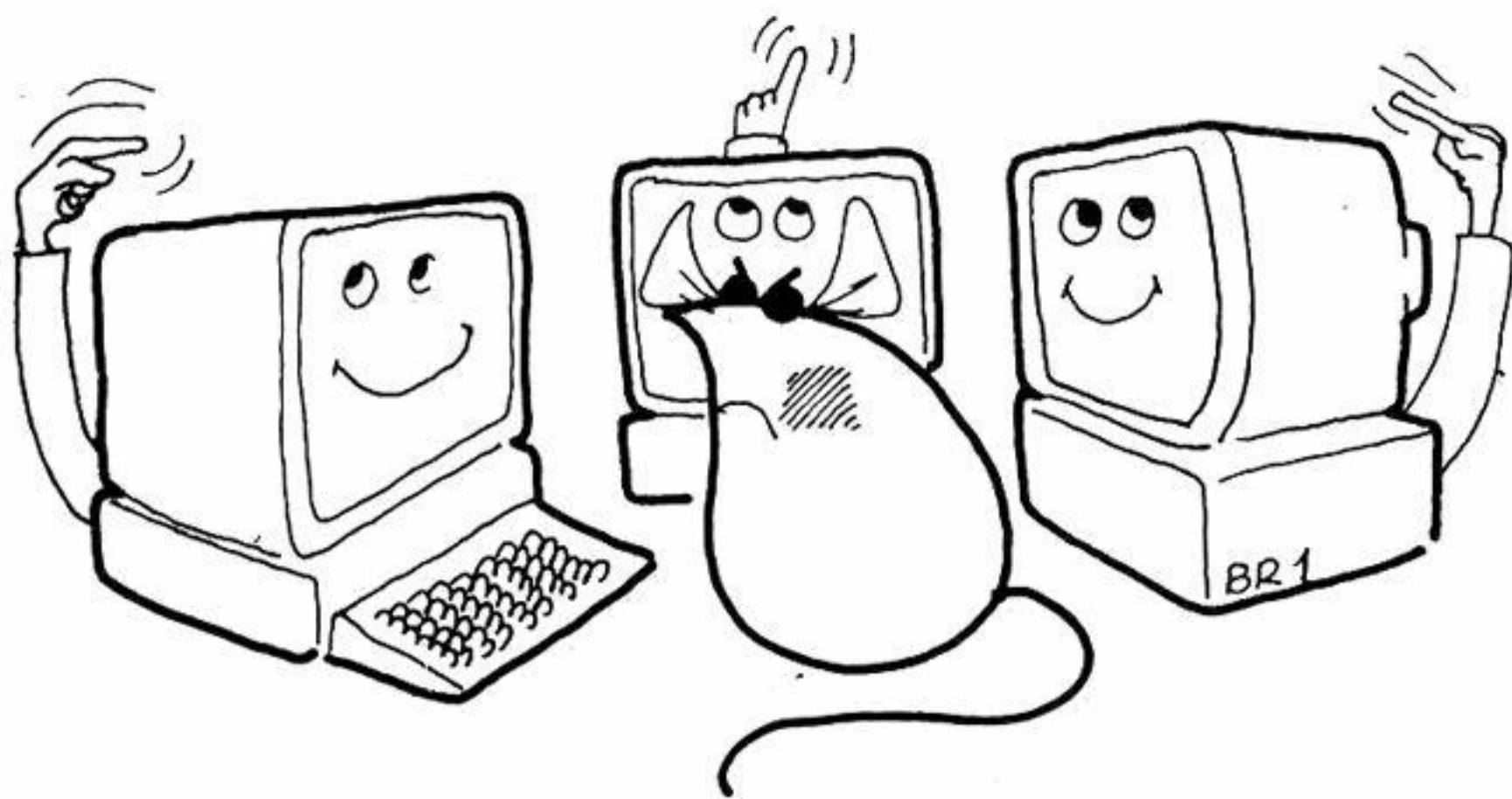
- sono disponibili nuovi comandi, in particolare delle comode funzioni di conversione: scrivendo, ad esempio...
decimal 100
...(senza Print nè parentesi) seguito dal tasto return, si ottiene 256; scrivendo "hex 160" si ottiene 00A0;
- il comando Dump invia alla stampante il listato del programma; Put, invece, lo invia al disco in formato SEQ (è in tale formato che vanno registrati i programmi Pascal); naturalmente il comando Load non riesce a trovare tali file sul disco, occorre caricarli tramite l'apposito comando Get; il comando Load "$",8 permette di caricare la Directory del disco e, come al solito, distrugge un eventuale programma in memoria.
- il comando Upper riporta la scrittura nel modo standard maiuscole - grafica; Lower fa tornare in minuscole - maiuscole; lo stesso effetto si può ottenere ancora premendo contemporaneamente i tasti Shift e CBM; lavorando in Pascal è comunque più comodo, come si vedrà, restare in modo Lower;
- il comando List opera in due modi diversi: se è preceduto dal carattere di doppio punto (:) funziona come di consueto; privo del doppio punto, invece, la pressione di un qualunque tasto arresta temporaneamente la visualizzazione, permettendo così di esaminare con calma il listato; per farlo riprendere basta premere un tasto qualunque.

Sempre a proposito di listati, il comando Dump serve per stampare il listing; purtroppo alcuni caratteri (quali le parentesi quadre e la freccia in alto) vengono "maltrattati" da tale comando; un listato più leggibile si ottiene con i comandi già noti ai nostri lettori...

Open 4,4,7: Cmd 4: List

...seguiti da Print#4: Close 4; attenzione a non abbreviare il Print con il punto di domanda.

- infine, per tornare al Basic, basta eseguire il comando Cold.

Come ulteriori differenze tra l'editor del Basic e quello del Pascal segnaliamo che quest'ultimo ha inserita la numerazione automatica (con step 10); e che eventuali spazi iniziali immessi tra il numero di linea ed il testo non vengono soppressi, a differenza di quanto accade con l'editor del Basic; si tratta di una possibilità molto comoda per ottenere listati più leggibili; peraltro i numeri di linea sono fittizi: l'esperto che volesse "spulciare" il contenuto dei file su disco, si accorgerebbe che questi non vengono registrati; ogni volta che un programma viene ricaricato da disco la numerazione è ricostruita ex novo, partendo dal numero 1000 con incrementi di 10.

LAVORANDO CON OXFORD PASCAL

Quando un programma è in memoria, può essere mandato in esecuzione col comando Run (abbreviabile con la sola R) che innesca, preliminarmente, la "compilazione" del programma, inesistente in Basic. Il programma, in altre parole, viene trascritto, in un'altra zona di memoria, in una forma più "comoda" per la successiva elaborazione.

Solo se questa fase ha buon esito, prenderà il via la "esecuzione" del programma stesso. In realtà le due fasi (compilazione ed esecuzione) possono essere separate: il comando "L" avvia la sola compilazione del programma e, se tale fase è andata in porto, un successivo comando "R" lo eseguirà.

Durante la fase di compilazione, innescata tramite L, l'intero programma riappare sul video (ovvero sulla stampante se, invece di L, si digita "P"), ed eventuali errori vengono segnalati a mano a mano che sono riscontrati; la segnalazione è effettuata in lettere maiuscole, ed è perciò più visibile se, come suggerito in precedenza, il programma è stato scritto in modo Lower; se si lavora direttamente tramite R, il programma non viene listato, e la segnalazione è fornita in modo leggermente diverso. Al termine dell'operazione appare un messaggio che riporta il numero totale di errori riscontrati (zero, ovviamente, se tutto è andato liscio...).

Il computer si dovrebbe rifiutare di dare avvio alla fase di esecuzione se la compilazione ha dato luogo ad errori, o se è stata alterata da successive operazioni impartite, ad esempio, in modo diretto.

Un semplice comando quale Let X = 3, dato dopo la fase L, obbliga il Computer a effettuare nuovamente la compilazione! In realtà un misterioso bug fa in modo che talvolta il Computer "dimentichi" di non aver eseguito la compilazione; l'esecuzione diretta di R, pertanto, può provocare dei guai, fino ad un blocco totale del sistema; per tale motivo è bene far sempre precedere il comando R dal comando L.

Se, in fase di compilazione, vengono segnalati errori, dobbiamo listare le linee incriminate e correggerle; poi ripetere i comandi L e R. Per quanto concerne la fase di revisione e correzione si tenga presente che possibilità interessanti sono offerte dai comandi NUMBER (per rinumerare il programma), FIND (per cercare delle stringhe), CHANGE (per sostituire una stringa con un'altra), DELETE (per eliminare linee dal programma). La numerazione automatica può essere disinserita digitando AUTO, e reinserita digitando AUTO N (ad esempio AUTO 50 se si vogliono incrementi di 50 in 50 per i numeri di linea); in fase di correzione degli errori, è comunque bene disabilitare la numerazione automatica: si eviterà, in tal modo, di distruggere inavvertitamente alcune linee, battendo return a vuoto sul numero di linea che il computer ha preparato.

PIOGGIA DI CORIANDOLI

```
procedure down;
(* in riga seguente mettere 40 spazi tra i due apici *)
     const blanc='
     a=$d800;b=$0400;
     type riga = packed array[1..40] of char;
          resto = packed array[1..960] of char;
 var y:resto;p1,p2:↑resto;p3:↑riga;
    procedure swap(x:integer);
     begin origin(p1,x+960);origin(p2,x+1000);origin(p3,x+40);
     y:=p1↑;p2↑:=y;p3↑:=blanc end;
begin swap(a);swap(b) end;
(* demo *)
begin border(3);pen(1);screen(0);
repeat down;pen(random);vdu(0,5*random div 32,'*') until false end.
```

La procedura DOWN, utilizzata nel programma dimostrativo, provoca lo scrolling verso il basso (una riga alla volta) del video; essa è scritta in modo da poter essere incorporata, così come è, in un qualunque altro programma. Il demo aggiunge sulla prima riga di schermo alcuni caratteri asterisco (*), generati casualmente e con colore casuale, che "cadono" ad una velocità notevole. Si tenga presente che, se basta far scrollare solo i caratteri dello schermo, e non anche i colori, si può sopprimere il comando SWAP(A) consentendo il raddoppio della velocità: l'esecuzione di DOWN avviene, infatii, in circa 0.04 secondi!

I DUE MODI DI OXFORD PASCAL

In realtà quello che abbiamo visto fin qui è il cosiddetto "modo residente" per lavorare con l'Oxford Pascal; eseguendo DISK si passa al "modo disco" (il comando inverso è RESIDENT), nel quale le operazioni di compilazione vengono svolte (tramite COMP) a partire da un file già presente su disco e forniscono un nuovo file, ancora su disco, pronto per l'esecuzione (comando EX) o per la "traduzione in Basic" (comando LOCATE, che genera un file di tipo programma, caricabile ed eseguibile normalmente da Basic).

Purtroppo, però, lavorando in modo disco, si devono fare i conti con l'esasperante lentezza del drive; quindi, per programmi non molto lunghi, conviene lavorare in modo residente. Se occorre, si utilizzeranno poi i comandi COMP, EX e LOCATE solo quando il programma è stato verificato e controllato.

ESTENSIONI DI OXFORD PASCAL DEDICATE AL C/64

Nell'Oxford Pascal non è prevista (ma è facilmente realizzabile) la gestione degli sprite. Sono invece già implementate le parole ENVEL, VOICE, VOLUME per la gestione del suono; BORDER, SCREEN, PEN per la gestione dei colori in bassa risoluzione; VDU per realizzare un velocissimo "Print AT"; HIRES, PAPER, INK, EXAMINE, per l'alta risoluzione; WINDOW permette di gestire lo schermo in forma mista alta-bassa risoluzione.

Sempre per l'alta risoluzione, è presente un mega-comando PLOT che, a seconda dei parametri forniti, svolge una qualunque delle funzioni SCRNCLR, DRAW, PAINT. Sono anche presenti, sia pure con nomi e/o sintassi un po' diverse, le funzioni e le operazioni che, in Basic, si realizzano tramite GET, PEEK, POKE, RND, SYS, TI$; come novità, rispetto al Basic, segnaliamo la presenza di IOTRAP, IOERROR (per la gestione degli errori di Input - Output); infine si tenga presente che il tasto STOP arresta l'esecuzione di un programma; se l'arresto avviene con schermo in alta risoluzione, occorre digitare (alla cieca) il comando KILL per tornare allo schermo normale.

LA MANUALISTICA

Due parole sulla documentazione sono doverose. Anzitutto, non sembra che esista un importatore ufficiale e le copie che circolano tra gli appassionati bisogna prenderle così come sono. Il manuale (in italiano) a corredo della versione giunta, per caso, in nostro possesso è purtroppo mal tradotto e zeppo di errori di stampa. In particolare, in certi programmi di esempio, la soppressione del simbolo piuttosto frequente di "freccia in alto" (presente tra il tasto Restore e l'asterisco) ed il sistematico arrotondamento delle parentesi (sono, cioè, riportate le tonde quando, al contrario, occorrerebbero le quadre) provocano non pochi guai all'utente poco esperto e impediscono a moltissimi programmi, se trascritti esattamente come sono, di girare.

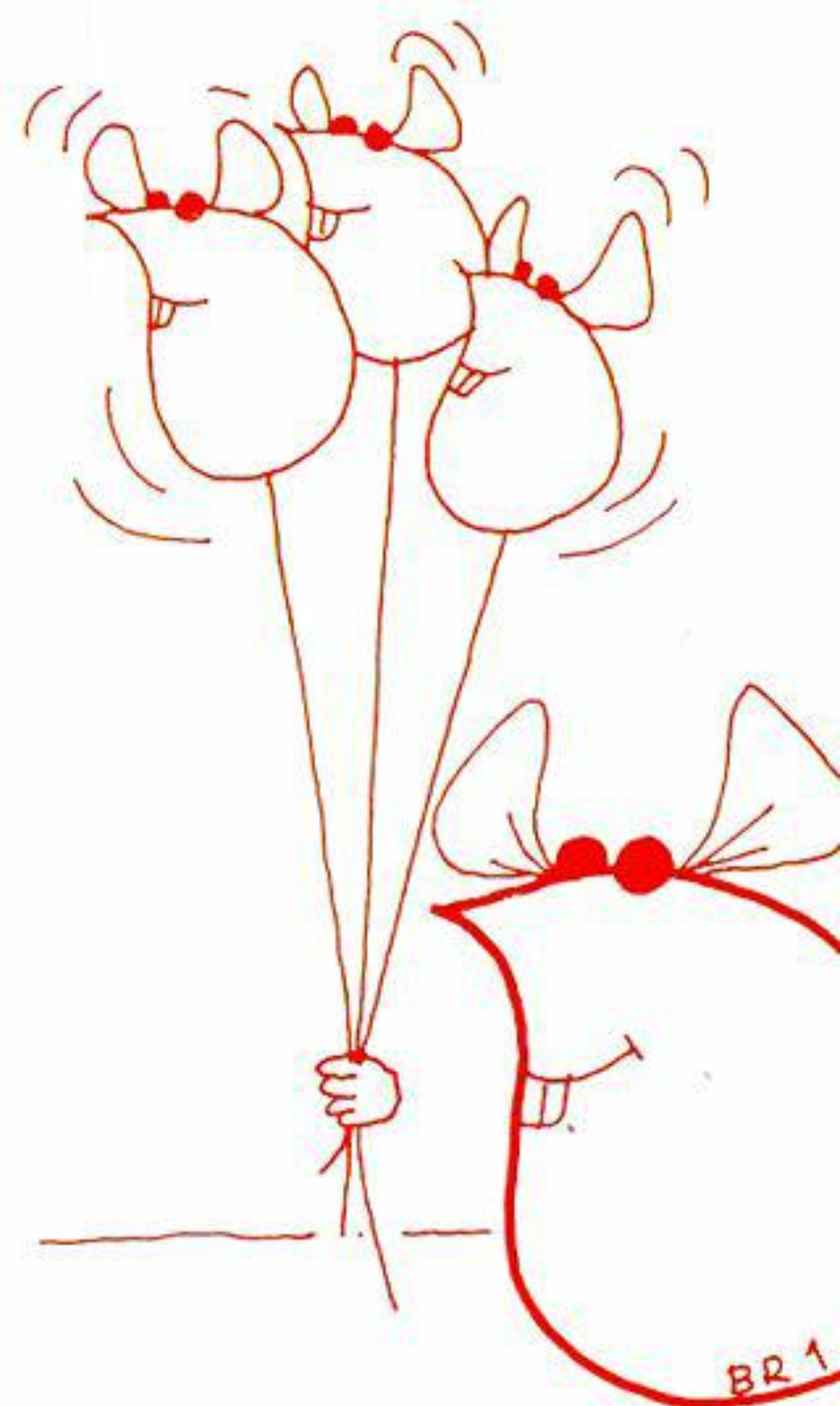
Per un uso corretto del simbolo "freccia in alto" si veda il programma dimostrativo di queste pagine, che fornisce una (velocissima) routine di scrolling verso il basso (Pioggia di coriandoli).

Per quanto riguarda lo sfruttamento delle risorse del C/64, si veda il riquadro specifico; a parte l'assenza di comandi idonei per la gestione degli sprite (gestione che, comunque, risulta facilmente programmabile) è da lamentare una scarsa sfruttabilità della grafica fine: non è infatti prevista la gestione della pagina grafica in modo multicolor e, nel modo alta risoluzione, i comandi tipo DRAW e PAINT operano solo su di una pagina a larghezza ridotta: 256 pixel in orizzontale, invece dei 320 usuali.

Per quanto riguarda, invece, la struttura generale, l'Oxford Pascal risulta un linguaggio sufficientemente completo. A prima vista le uniche due assenze di rilievo concernono il tipo STRING, ammesso solo nelle costanti; volendo usare stringhe variabili si può definirle come PACKED ARRAY OF CHAR, ma bisogna poi costruirsi tutto: dalla lettura alla scrittura, dalla concatenazione al confronto...

Altra carenza significativa è la mancanza del tipo LONG-INTEGER (anche qui: definirlo è banale, ma la costruzione di funzioni di I/O, operazioni e confronti è abbastanza lunga). La mancanza del tipo LONG-INTEGER si rivela, in particolare, piuttosto fastidiosa anche perchè, in carattere con lo spirito del Pascal, la comodissima funzione che in Basic si scrive INT non esiste; ed i suoi sostituti ROUND e TRUNC forniscono errore se applicate a numeri che, in valore assoluto, superano 32767.

Per ovviare a tale mancanza si può fare uso delle funzioni e procedure presentate nel programma di queste pagine, che permette di lavorare con numeri che, in valore assoluto, non superino 2147483647; la lunghezza del programma è giustificata, oltre che dal guadagno sensibile nel campo numerico a disposizione, dai numerosi problemi causati da comportamenti un po' troppo... "coerenti" dell'Oxford Pascal". Per rendersi conto di che sorprese può riservare tale coerenza dovrebbe bastare dare un'occhiata ai tre esempi seguenti:

ESEMPIO 1
I comandi...
WRITE(1E5:1:4)

...e...

WRITE(1E8:1:0)

...che dovrebbero far scrivere i numeri 100 mila e 100 milioni con, rispettivamente, quattro e zero cifre dopo la virgola, generano gli output 099999.9998 e 099999999.8

ESEMPIO 2

Il mini-programma...

100 var x:real; y:integer;
110 begin y:=30000;
120 x:=y*y;
130 writeln(x);end.

...genera un fastidiosissimo INTEGER OVERFLOW, LINE 120; se in linea 110 si sostituisce il valore 30000 col valore 65535 il programma "gira", ma il valore di X è... 1 (perchè Y vale -1)

ESEMPIO 3

Come è noto fin dai tempi degli antichi greci, l'operazione di "divisione con resto" è uno strumento molto potente nell'ambito dei numeri interi; il linguaggio Pascal dispone allo scopo delle operazioni DIV e MOD che forniscono, rispettivamente, il quoziente intero ed il resto della divisione. Come applicazione standard di tali operatori (e della ricorsività permessa dal Pascal) si ha il seguente programma, che chiede due numeri e ne fornisce il Massimo Comun Divisore, secondo il ben noto "algoritmo di Euclide":

100 var a,b:integer;
110 function mcd(u,v:integer): integer;
120 begin if v=0 then mcd: =uelse mcd:=mcd(v,u mod v) end;
130 begin read(a,b); write(mcd(a,b):0) end.

E' un mini-programma elegante e velocissimo, ma funziona solo se i due numeri inseriti sono compresi tra 0 e 32767; il lettore è invitato a scoprire da solo i pasticci che avvengono con numeri fuori di tale intervallo, ed i motivi per cui, tutto sommato, nonostante i risultati forniti sembrino strani, da un linguaggio preciso come il Pascal ci si dovevano aspettare certe sorprese.

UN PROGRAMMA PER GESTIRE GLI INTERI-LUNGHI

```
function int(a:real):real;const base-2030;max-2147483647.;
   var x,y,z,v,w:integer;b:real;r:↑real;
   begin if abs(a)>max then int:-a
      else begin origin(r,base);r↑:-abs(a);
            y:-peek(base-6)-128;w:-255;
            if y<-0 then poke(base-6,0);
            for x:-base-5 to base-2 do begin z:-andb(w,peek(x));
               if y>7 then y:-y-8
               else begin w:-0;v:-shl(255,8-y);poke(x,andb(z,v)) end
            end;
            b:-r↑;if a<0 then begin b:--b;if abs(a)+b>0 then b:-b-1 end;
            int:-b
            end
   end;
function longint(a:real):boolean;const max-2147483647.;
   begin if abs(a)>max then longint:-false else longint:-(a-int(a))
   end;
procedure leggi(var a:real);var b:integer;
   begin readln(a);if not longint(a) then b:-1 div 0
   end;
function resto(a,b:real):real;
   begin if b<0 then begin a:--a;b:--b end;
   if b-0 then resto:-a else resto:-a-b*int(a/b)
   end;
procedure scrivi(a:real);var x,z:integer;b:packed array[0..10] of integer;
   begin if not longint(a) then write(a)
     else begin if a<0 then begin write('-');a:--a end;
          x:--1;
          repeat
          x:-x+1;b[x]:-trunc(resto(a,10));a:-(a-b[x])/10)
          until a-0;
          for z:-x downto 0 do write(chr(48+b[z]))
          end
```

Il programma permette un trattamento accurato di "interi lunghi", cioè di numeri interi che, in valore assoluto, sono inferiori a 2 exp 31; si faccia attenzione, nel ricopiarla, a non scordare il punto finale dopo il valore della costante MAX.

A prezzo di qualche ripetizione (ad esempio la costante MAX figura sia nel corpo della funzione INT sia in quello della funzione LONGINT) le routine sono autosufficienti, e possono essere incorporate in un qualunque altro programma; per il loro uso si tenga presente che:

• la funzione INT(X) richiede X variabile reale ed opera esattamente come la funzione INT del Basic; la sua definizione è un po' "sporca", ed antitetica allo "spirito pascaliano"; però funziona...

• la funzione LONGINT(X) fornisce, per X variabile reale, un flag col valore TRUE se X è un intero lungo, e col valore FALSE se X è troppo grande o non intero.

• la procedura LEGGI attende l'introduzione di un numero intero lungo, e segnala errore (INTEGER OVERFLOW) se il valore introdotto non rispetta la condizione.

• la procedura SCRIVI(X) fornisce la scrittura corretta di tutte le cifre della variabile reale X se il valore di X è un intero lungo; in caso contrario opera esattamente come un WRITE(X).

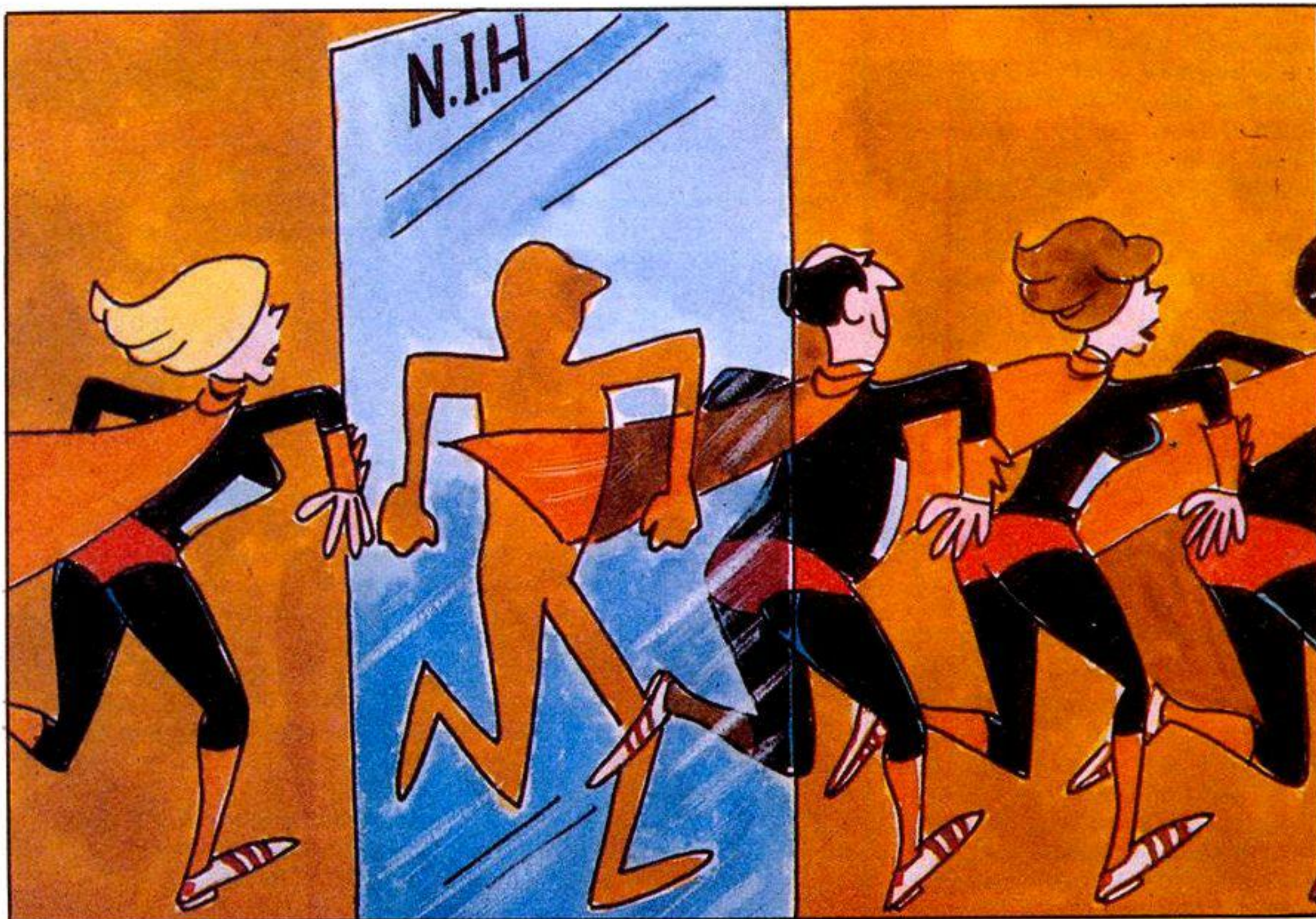
• infine la funzione RESTO(A,B) opera esattamente come la primitiva A MOD B, ma permette agli argomenti A e B di essere interi lunghi.

Per un DEMO si premetta alle routine precedenti la linea:

var u,v,x,y,z:real;

e si aggiungano in coda le linee:

```
function mcd(a,b:real):real;
   begin if b-0 then mcd:-a else mcd:-mcd(b,resto(a,b))
   end;
begin write('primo numero - ');leggi(x);
      write('altro numero - ');leggi(y);
      z:-mcd(abs(x),abs(y));write('mcd - ');scrivi(z);writeln;
      write('mcm - ');if z-0 then writeln(' 0')
            else begin u:-x/z;v:-y/z;
                 scrivi(x*v);writeln;write('fattori : ');scrivi(u);
                 write(' e ');scrivi(v);writeln
                 end
end.
```

Si ha così la versione per interi lunghi dell'algoritmo di Euclide: il programma chiede due numeri e, se ottiene due interi lunghi, ne calcola MCD e mcm.

UNO SPRITE PER LE TUE SCELTE

...e un po' di teoria per conoscere meglio il Commodore 64

di **Valentino Spataro**

Gli appassionati di musica e di fotografia hanno una quantità enorme di cassette e diapositive.

Giustificato, quindi, sarà l'uso del computer per memorizzare tutte le notizie possibili, per trovare velocemente quella vecchia cassetta che piace tanto alla mamma o quella diapositiva, da mostrare agli amici, in cui fa la sua bella figura una ragazza conosciuta al mare.

Con il tempo, però, stanchi del vecchio sistema di scelta delle opzioni (ad ognuna di esse corrisponde un tasto) e considerato lo sviluppo delle icone come mezzo di comunicazione con il computer, abbiamo deciso di applicare le nuove idee al vecchio archivio.

Se non vi interessa disporre di un sistema di puntamento come (e meglio) di un mouse per il vostro C/64, ma avete solamente voglia di imparare qualche tecnica nuova, il consiglio è di proseguire nella lettura dell'articolo in cui verrano descritte, nel modo più semplice, le tecniche usate.

A CHE SERVONO LE ROUTINE

Le routine proposte sono due: una impiega il joystick, l'altra la tastiera; la loro funzione è di spostare uno sprite attraverso lo schermo. Inoltre, premendo Fire (o CTRL per la versione per tastiera), verrà "cattura-

to" il carattere presente sotto l'angolo superiore SINISTRO dello sprite n. 0 (che potrà coincidere, ad esempio, con la punta di una freccia).

Il movimento dello sprite, nella versione per joystick, si ottiene inserendo questo strumento in porta 2. La versione per tastiera richiede la pressione contemporanea del tasto Commodore per l'attivazione; tenendo premuto quel tasto, e servendosi dei tasti cursore (e, come vedremo, del tasto CTRL) muoverete lo sprite senza esser costretti a spostare, contemporaneamente, l'eventuale cursore sullo schermo.

Grazie a questo sistema sarà quindi possibile scegliere un'opzione semplicemente spostando lo sprite sull'indicazione del tasto da premere; facciamo un esempio.

Supponiamo di avere il seguente programma:

```
10 For i=1 to 10: Print i, no$(i): Next
20 Input "filename "; fl$: rem chiede il nome del programma da caricare
30 load fl$: rem carica il file
```

...che visualizza il vettore dei 10 nomi tra cui scegliere il file da caricare. Questa operazione è spesso fastidiosa, specialmente se il nome del file (come spesso capita) è una sequenza di caratteri insoliti.

Basterà, allora, attivare in precedenza una delle due routine pubblicate e far passare lo sprite sulla scritta desiderata tenendo premuto Fire (oppure CTRL e Commodore a seconda della versione digitata).

Ricordate che, in genere, la routine legge solo quello che c'è sullo schermo: per "confermare" la scelta sarà necessario usare il tasto return.

In ogni caso è sempre possibile memorizzare, in una variabile stringa, la scritta "prelevata" dallo schermo.

Per attivare le routine dovete trascrivere le linee di Data e usare, come caricatore, un programmino qualsiasi tenendo conto dello standard della presente rubrica.

CARATTERISTICHE DELLE ROUTINE

Le routine provvedono a cambiare i contenuti della locazione 53248, 53248+1 e 53248+16 che indicano, al computer, le coordinate dello sprite n. 0; per tutto il resto si dovrà procedere da Basic, sia per l'attivazione dello sprite, sia per il colore, sia per l'espansione delle dimensioni.

Le routine, ovviamente rilocabili, lavorano contemporaneamente al programma Basic già in esecuzione. La versione per joystick lavora in interrupt (piccola deroga alle rigide norme dell'enciclopedia), mentre la versione per tastiera si inserisce nella routine addetta alla scansione della stessa tastiera, quella cioè richiamata ogni volta che si preme un tasto.

Le routine, oltre a cambiare le coordinate dello sprite, provvedono a calcolare le coordinate dello schermo. Per determinarle a parte, basteranno queste semplici linee basic:

```
10 cellavideo = peek (254) + peek (255) * 256 +
   + peek (782)
20 y = int(cellavideo - 1024) / 40
30 x = (cellavideo - 1024) - y * 40
```

Una volta calcolata su quale locazione dello schermo si trova lo sprite, il programma ne può leggere il contenuto; il valore trovato, però, è in formato "poke": bisogna quindi convertirlo in formato Ascii (come potrete vedere dal disassemblato) prima di metterlo nel buffer di tastiera.

Il calcolo delle coordinate dello schermo, e della conversione in formato Ascii, sono due procedure identiche, come potrete notare nel disassemblato e nelle linee DATA: se, quindi, vorrete trascrivere entrambe le routine, riutilizzate pure le linee data uguali, ma attenzione agli ultimi 4 dati dell'ultima riga.

Infine ecco l'elenco delle locazioni usate dal programma: oltre alle 254 e 255 anche la 253 e la 168. Inoltre la locazione 167 contiene la velocità dello sprite (regolabile) e la 169 la velocità di acquisizione dei dati dallo schermo. Le locazioni 167, 168 e 169 sono usate, di solito, solamente nel caso si usi il registratore. Ogni volta che si finisce di usarlo, bisognerà reimpostare i valori che consigliamo qui di seguito.

Per la versione tastiera le locazioni 197 e 199 dovranno contenere, rispettivamente, i valori (7, 0), (3, 2), (1, 4), (0, 8) per passare dalla massima alla minima velocità.

Per la versione joystick, invece, la locazione 199 conterrà sempre il valore 9, mentre la 197 avrà i valori 7, 5, 3, 1, 0 a mano a mano che si intenda passare dalla massima alla minima velocità.

Sono state scelte tali locazioni perchè, generalmente, sono poco usate e questo fatto garantisce un'elevata compatibilità con molti programmi.

Inoltre le routine provvedono a mantenere lo sprite entro i limiti dello schermo; tuttavia, conoscendo lo start address della routine, potete cambiarne i valori:

Versione per:

tastierajoy
 146138 :xmin
 152144 :xmax
 8785 :ymin
 9973 :ymax

Basterà quindi un semplice POKE SA+87,150 per far sì che nella versione per tastiera lo sprite non possa mai andare nella metà superiore dello schermo.

Volendo individuare, da Basic, la pressione del tasto fire, dovrete scrivere, ad esempio:

10 if peek(56320) = 111 then print "tasto fire premuto"

Per la pressione del tasto CTRL, invece:

10 if peek(653) = 4 then print "tasto CTRL premuto"

Raccomandiamo di alterare il Top di memoria nel caso in cui si decida di allocare la routine in un'area gestita automaticamente dal Basic. Si tenga presente, infatti, che la tecnica prevede l'inserimento nel buffer di tastiera del carattere presente sotto lo sprite. La garbage collection, quindi, può risultare più attiva del solito.

Ora qualche considerazione per la versione per tastiera: premendo il tasto Commodore si abilitano solo i tasti cursore, lo SHIFT e CTRL. Durante lo spostamento dello sprite il cursore deve stare fermo: a questo provvedono le linee 100 e 101 del disassemblato relativo alla versione per tastiera.

Per demo potrete usare il banale programmino proposto. I disassemblati non sono commentati, ma sono state usate sigle più semplici di tante parole.

LA ROUTINE DI SCANSIONE DELLA TASTIERA

Questa routine viene richiamata ogni volta che si preme un tasto. 655 e 656 sono i suoi puntatori che indicano, di solito, la procedura posta da 60232 ($EB48). Cambiando i valori dei puntatori possiamo inserire una nostra routine, purchè sia sempre presente, alla sua fine, un JMP $EB48.

I registri interessati sono:

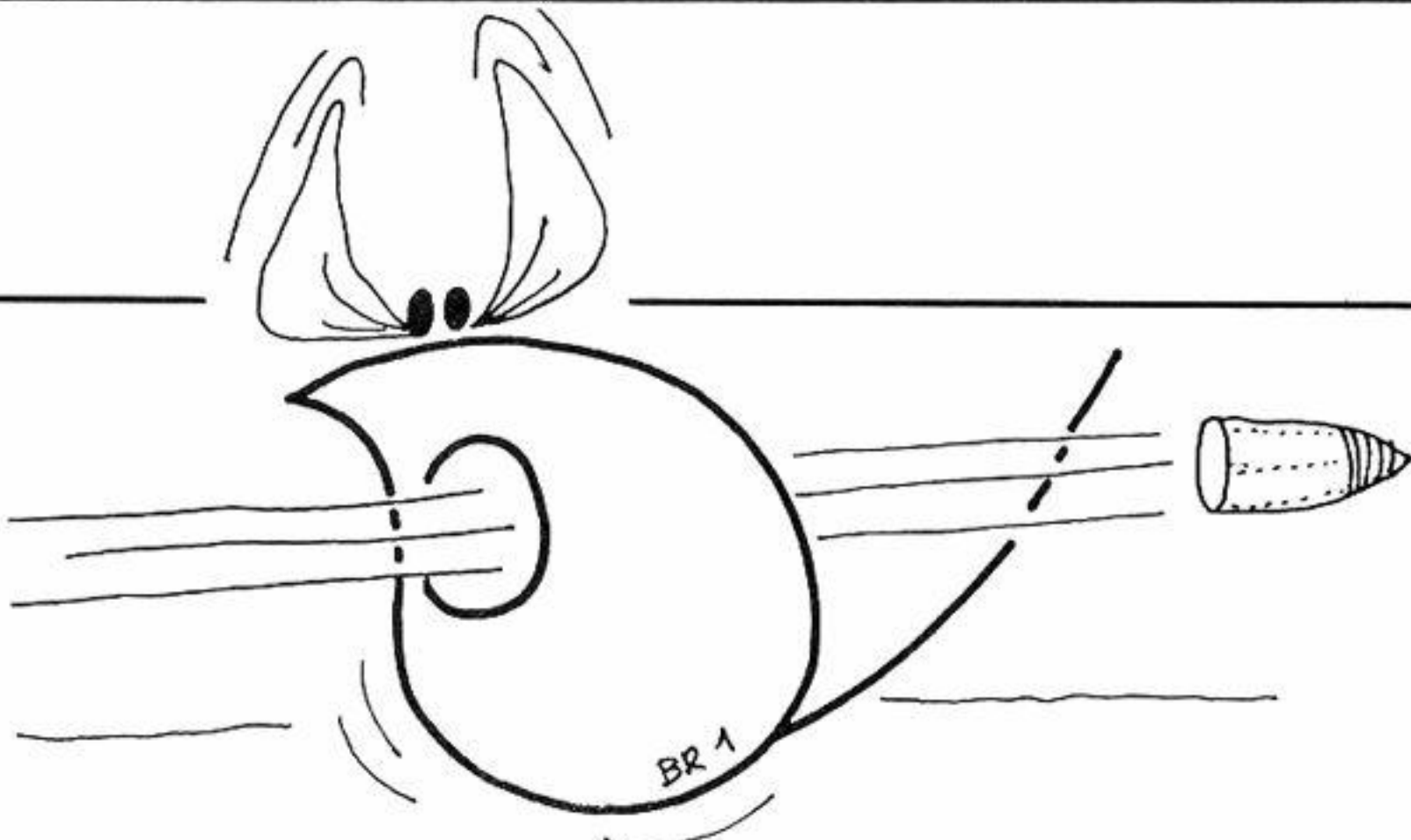
- 655 - 656 puntatori a routine scansione;
- 653 indica se sono premuti i tasti CTRL, SHIFT, Commodore;
- 650 flag per ripetizione dei tasti:
  128 tutti tasti con repeat
  64 nessun tasto con repeat
  0 situazione standard;
- 631 - 640 buffer di tastiera: contiene fino a 10 tasti premuti, e non ancora utilizzati, in formato Ascii;
- 198 indica quanti tasti sono stati premuti (e quindi quanti se ne trovano in 631 - 640)
- 203 contiene il valore del tasto attualmente premuto. Il valore non è nè in formato Ascii nè Poke: usare il seguente programma per vederne i valori:

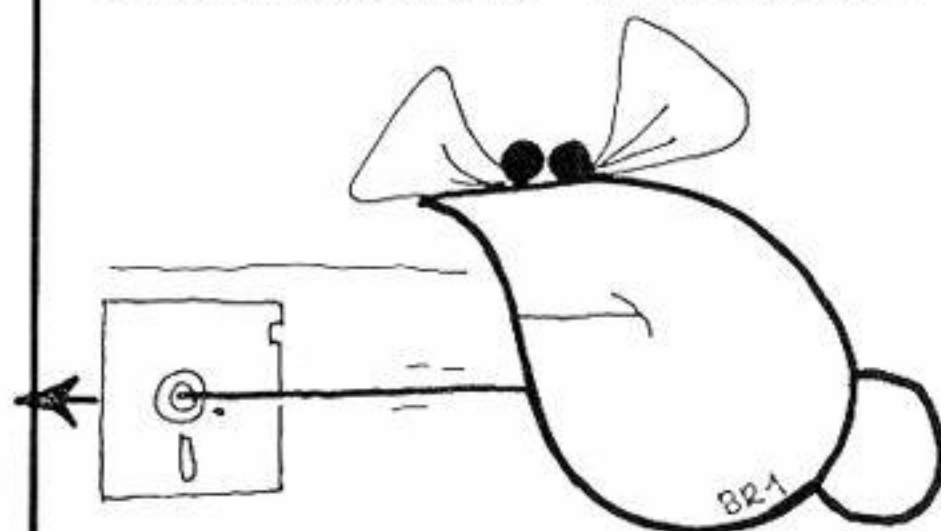
10 print chr$ (147) peek (203): goto 10

- 197 in questa locazione la routine di scansione di tastiera deposita il valore dell'ultimo tasto premuto. Da basic ha lo stesso valore di 203 ma non è un doppione: serve anche nel caso di ripetizione (650);

Da notare che, premendo insieme Commodore e CTRL, alcuni tasti prendono un altro codice: per questo, nel disassemblato della versione per tastiera, da 47 a 64 si valuta la pressione dei due tasti cursore (valori 2 e 5) ma anche quella del tasto F3 in quanto premendo Commodore con CTRL e CRSR destra / sinistra il valore contenuto in 203 è 7 e non 5 come dovrebbe essere, appunto, a F3.

```
1 ****************************
2 * POINTER BY TASTIERA
3 * PREMERE C64 E CRSR PER MOVE
   E CTRL PER FIRE *
4 ****************************
5             ORG     40000
6  SPEED      =       167
7  RIPETE1    =       168
8  RIPETE2    =       169
9  TPREMUTO   =       203
10 TPRECED    =       197
11 SHFTFLAG   =       653
12 TASTPOIN   =       655
13 YREC       =       53249
14 XREC       =       53248
15 XALTO      =       53248+16
16 ****************************
17 START      CLC
18            LDA     $14
19            ADC     #START1-START
20            STA     TASTPOIN
21            BCC     SALTO1
22            INC     $15
23 SALTO1     LDA     $15
24            STA     TASTPOIN+1
25            RTS
26 ****************************
27 * CONTROLLA PRESSIONE C=
28 ****************************
29 START1     LDA     SHFTFLAG
30            LSR
31            LSR
32            BCS     INIZIO
33            JMP     $EB48
34 INIZIO     LDY     SPEED
35 COMPARA    CPY     #$FF
36            BNE     PARTENZA
37            SEC
38            BCS     HELP
39 ****************************
40 * ANALISI PRESSIONE TASTI
41 ****************************
42 PARTENZA   LDA     SHFTFLAG
43            LSR
44            LSR
45            LSR
46            BCS     BEGIN
47 BEGIN      LDX     TPREMUTO
48            LDA     SHFTFLAG
49            LSR
50            BCS     SISHIFT
51            CPX     #2
52            BEQ     DESTRA
53            CPX     #5
54            BEQ     DESTRA
55            CPX     #7
56            BEQ     BASSO
57            BNE     FIRE
58 SISHIFT    CPX     #2
59            BEQ     SINISTRA
60            CPX     #5
61            BEQ     SINISTRA
62            CPX     #7
63            BEQ     ALTO
64            BNE     FIRE
65 PART1      BNE     PARTENZA
66 ALTO       LDA     YREC
67            CMP     #51
68            BCC     FINE
69            DEC     YREC
70 HELP       BCS     FINE
71 BASSO      LDA     YREC
72            CMP     #249
73            BCS     FINE
74            INC     YREC
75            BCC     FINE
76 DESTRA     INC     XREC
77            BNE     LOOP3
78            LDA     XALTO
79            ORA     #1
80            STA     XALTO
81            CLC
82            BCC     LOOP3
83 SINISTRA   DEC     XREC
84            BPL     LOOP3
85            LDA     XALTO
86            AND     #$FE
87            STA     XALTO
88 LOOP3      LDA     XALTO
89            LSR
90            LDA     XREC
91            BCS     LOOP4
92            CMP     #25
93            BCC     DESTRA
94            BCS     FINE
95 LOOP4      CMP     #85
96            BCS     SINISTRA
97 FINE       DEY
98            CPY     #$FF
99            BNE     PART1
100           LDA     #64
101           STA     TPREMUTO
102 FIRE      LDA     SHFTFLAG
103           LSR
104           LSR.
105           LSR
106           BCS     PRENDE
107 EXIT      JMP     $EB48
108 PRENDE    INC     RIPETE1
109           LDA     RIPETE1
110           CMP     RIPETE2
111           BMI     EXIT
112           LDA     #0
113           STA     RIPETE1
114 ***********************
115 * PARTE COMUNE
116 ***********************
```

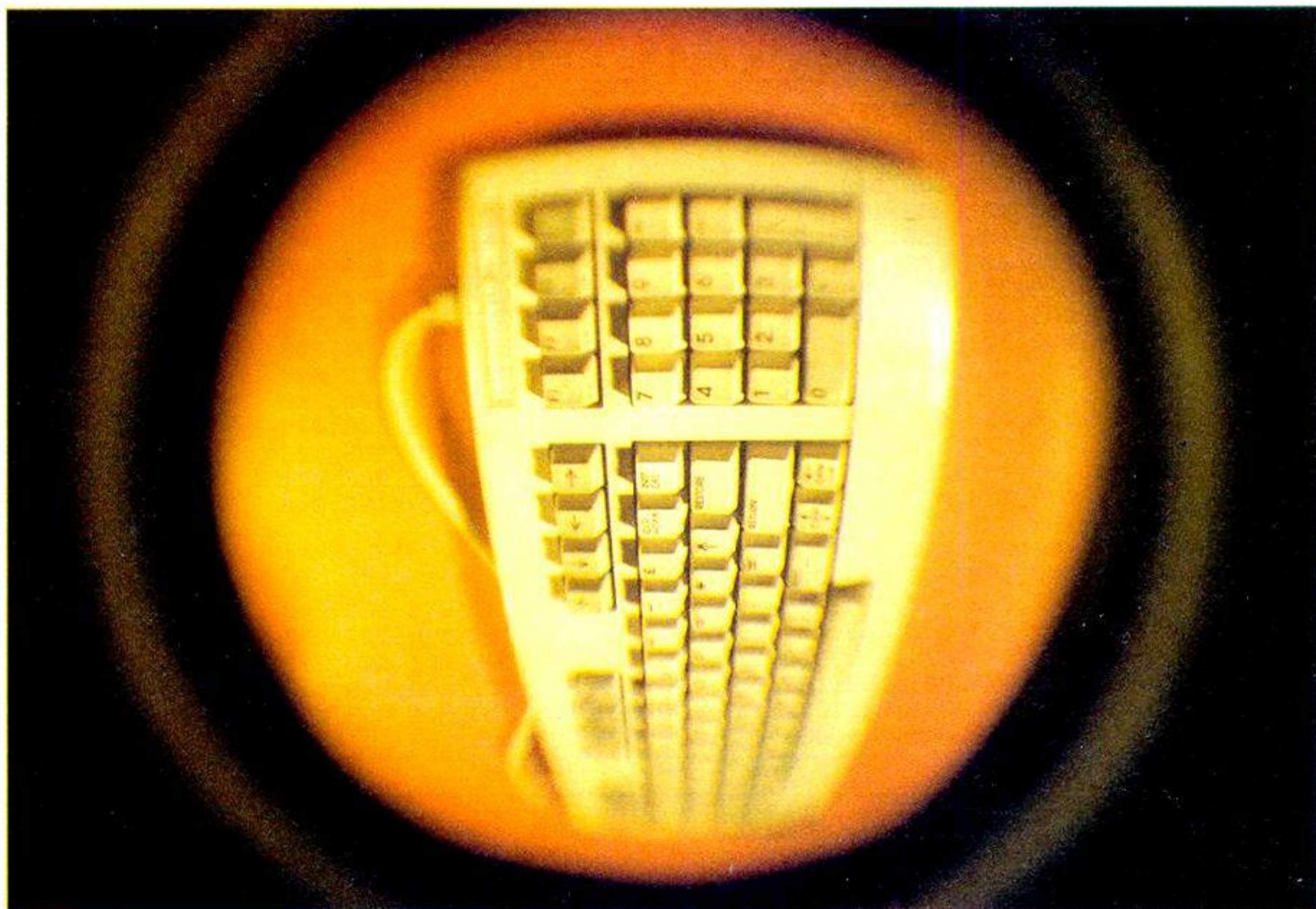
```
1 ***************************
2 * INTERRUPT & JOY
3 ***************************
4             ORG     40000
5  RIPETE1    =       168
6  RIPETE2    =       169
7  SPEED      =       167
8  XREC       =       $D000
9  YREC       =       $D001
10 XALTO      =       $D010
11 INPUTJOY   =       $DC00
12 CINV       =       $0314
13 ***************************
14 START      SEI
15            CLC
16            LDA     $14
17            ADC     #START1-START
18            STA     CINV
19            BCC     SALTO1
20            INC     $15
21 SALTO1     LDA     $15
22            STA     CINV+1
23            CLI
24            RTS
25 **********************
26 LOOP10     LDA     INPUTJOY
27            LSR
28            LSR
29            LSR
30            LSR
31            LSR
32            BCC     LOOP11
33            LDA     #00
34            STA     RIPETE1
35            BEQ     BEGIN
36 LOOP11     INC     RIPETE1
37            LDA     RIPETE1
38            CMP     RIPETE2
39            BNE     BEGIN
40            LDA     #0
41            STA     RIPETE1
42            BEQ     FIRE
43 START1     LDY     SPEED
44 REPEAT     CPY     #$FF
45            BNE     LOOP10
46            JMP     $EA31
47 BEGIN      LDA     INPUTJOY
48            LSR
49            TAX
50            BCC     ALTO
51            LSR
52            BCS     LOOP2
53 BASSO      LDA     YREC
54            CMP     #249
55            BCS     LOOP2
56            INC     YREC
57            BNE     LOOP2
58 ALTO       LDA     YREC
59            CMP     #51
60            BCC     LOOP2
61            DEC     YREC
62 LOOP2      TXA
63            LSR
64            LSR
65            BCC     SINISTRA
66            LSR
67            BCS     END
68 DESTRA     INC     XREC
69            BNE     LOOP3
70            LDA     XALTO
71            ORA     #$01
72            STA     XALTO
73            CLC
74            BCC     LOOP3
75 SINISTRA   DEC     XREC
76            BPL     LOOP3
77            LDA     XALTO
78            AND     #$FE
79            STA     XALTO
80 LOOP3      LDA     XALTO
81            LSR
82            LDA     XREC
83            BCS     LOOP4
84            CMP     #25
85            BCC     DESTRA
86            BCS     END
87 LOOP4      CMP     #$55
88            BCS     SINISTRA
89 END        DEY
90            CLC
91            BCC     REPEAT
92 **************************
93 * PROCEDURA PER DETERMINARE
94 * LE COORDINATE IN LOW-RES
95 **************************
96 FIRE       LDA     #00
97            STA     $FD
```

```
98              LDA    #04
99              STA    $FE
100             LDA    YREC
101             SBC    #50
102             LSR
103             LSR
104             LSR
105             TAX
106 LOOP5       CPX    #00
107             BEQ    LOOP6
108             LDA    #40
109             CLC
110             ADC    $FD
111             STA    $FD
112             BCC    SALTO3
113             INC    $FE
114 SALTO3      DEX
115             CLC
116             BCC    LOOP5
117 LOOP6       LDA    XREC
118             LSR
119             LSR
120             LSR
121             TAY
122             LDA    XALTO
123             LSR
124             BCS    LOP1
125             TYA
126             SBC    #2
127             CLC
128             BCC    LOP2
129 LOP1        TYA
130             ADC    #28
131 LOP2        TAY
132             LDA    ($FD),Y
133 ******************************
134 * CONVERTE IL CONTENUTO DELLA
135 * CELLA DI MEMORIA DA FORMATO
136 * POKE IN FORMATO ASCII
137 ******************************
138             CMP    #128
139             BMI    S1
140             SBC    #128
141 S1          CMP    #95
142             BMI    S2
143             ADC    #63
144             CLC
145             BCC    METTE
146 S2          CMP    #64
147             BMI    S3
148             ADC    #31
149             CLC
150             BCC    METTE
151 S3          CMP    #32
152             BPL    METTE
153             ADC    #64
154 METTE       STA    631
155             LDA    #1
156             STA    198
157             JMP    $EA31
158 *********************
```

VUOI PUBBLICARE UN ANNUNCIO GRATUITO?

Commodore Computer Club ti offre la possibilità di pubblicare GRATUITAMENTE, e immediatamente dopo che lo spedisci, il tuo annuncio riguardo scambio o vendita di software, vendita o acquisto di apparecchi usati, ricerca di amici per fondare un club e così via. Per fare in modo che il tuo annuncio compaia nel prossimo numero della rivista siamo costretti a seguire una procedura rapidissima e dobbiamo quindi chiederti una piccola cortesia: compila le righe che seguono usando un grosso pennarello (o "ripassandole" più volte con la penna), tenendo presente che IL TUO STESSO ANNUNCIO verrà ridotto, per mezzo di procedimenti fotografici, a cm. 7 x 3 circa e pubblicato così come ci perviene in redazione, senza alcuna modifica. Ti consigliamo, pertanto, di usare una calligrafia chiara, a stampatello e dotata di caratteri sufficientemente grandi per consentire un'agevole lettura anche dopo il "trattamento" di riduzione.

Invia l'annuncio in busta chiusa, affrancata secondo le norme vigenti, indirizzando a:

Commodore Computer Club
Viale Famagosta, 75
20142 MILANO

```
1000 PRINTCHR$(147)"CON SYS XXXX LA ROUTINE PERMETTE DI
1010 PRINT"MUOVERE UNO SPRITE USANDO LA TASTIERA"
1020 PRINT"SENZA INTERFERIRE CON IL PROGRAMMA
1030 PRINT"CORRENTE (VEDI ARTICOLO)"
1040 REM ROUTINE PER TASTIERA
1050 DATA 024,165,020,105,018,141,143,002,144,002,230,021,165,021
1060 DATA 141,144,002,096,173,141,002,074,074,176,003,076,072,235
1070 DATA 164,167,192,255,208,003,056,176,056,173,141,002,074,074
1080 DATA 074,176,000,166,203,173,141,002,074,176,014,224,002,240
1090 DATA 050,224,005,240,046,224,007,240,030,208,097,224,002,240
1100 DATA 052,224,005,240,048,224,007,240,004,208,083,208,210,173
1110 DATA 001,208,201,051,144,065,206,001,208,176,060,173,001,208
1120 DATA 201,249,176,053,238,001,208,144,048,238,000,208,208,024
1130 DATA 173,016,208,009,001,141,016,208,024,144,013,206,000,208
1140 DATA 016,008,173,016,208,041,254,141,016,208,173,016,208,074
1150 DATA 173,000,208,176,006,201,025,144,214,176,004,201,085,176
1160 DATA 224,136,192,255,208,177,169,064,133,203,173,141,002,074
1170 DATA 074,074,176,003,076,072,235,230,168,165,168,197,169,048
1180 DATA 245,169,000,133,168
1190 REM PARTE COMUNE ALLE DUE ROUTINES
1200 DATA 169,000,133,253,169,004,133,254,173
1210 DATA 001,208,233,050,074,074,074,170,224,000,240,015,169,040
1220 DATA 024,101,253,133,253,144,002,230,254,202,024,144,237,173
1230 DATA 000,208,074,074,074,168,173,016,208,074,176,006,152,233
1240 DATA 002,024,144,003,152,105,028,168,177,253,201,128,048,002
1250 DATA 233,128,201,095,048,005,105,063,024,144,015,201,064,048
1260 DATA 005,105,031,024,144,006,201,032,016,002,105,064,141,119
1270 DATA 002,169,001,133,198,076,072,235,-1,34267
1280 END
```

```
1000 PRINTCHR$(147)"CON SYS XXXX LA ROUTINE PERMETTE DI
1010 PRINT"SPOSTARE LO SPRITE N.0 SULLO SCHERMO TRAMITE JOYSTICK E,
1020 PRINT"PREMENDO FIRE, DI PRELEVARE IL CARATTERE PRESENTE
1030 PRINT"SOTTO LO SPRITE E INSERIRLO NEL BUFFER
1040 PRINT"DI TASTIERA.
1050 DATA 120,024,165,020,105,050,141,020,003,144,002,230,021,165
1060 DATA 021,141,021,003,088,096,173,000,220,074,074,074,074,074
1070 DATA 144,006,169,000,133,168,240,023,230,168,165,168,197,169
1080 DATA 208,015,169,000,133,168,240,101,164,167,192,255,208,220
1090 DATA 076,049,234,173,000,220,074,170,144,015,074,176,022,173
1100 DATA 001,208,201,249,176,015,238,001,208,208,010,173,001,208
1110 DATA 201,051,144,003,206,001,208,138,074,074,144,019,074,176
1120 DATA 048,238,000,208,208,024,173,016,208,009,001,141,016,208
1130 DATA 024,144,013,206,000,208,016,008,173,016,208,041,254,141
1140 DATA 016,208,173,016,208,074,173,000,208,176,006,201,025,144
1150 DATA 214,176,004,201,085,176,224,136,024,144,157
1160 REM PARTE COMUNE ALLE DUE ROUTINES
1170 DATA 169,000,133,253,169,004,133,254,173
1180 DATA 001,208,233,050,074,074,074,170,224,000,240,015,169,040
1190 DATA 024,101,253,133,253,144,002,230,254,202,024,144,237,173
1200 DATA 000,208,074,074,074,168,173,016,208,074,176,006,152,233
1210 DATA 002,024,144,003,152,105,028,168,177,253,201,128,048,002
1220 DATA 233,128,201,095,048,005,105,063,024,144,015,201,064,048
1230 DATA 005,105,031,024,144,006,201,032,016,002,105,064,141,119
1240 DATA 002,169,001,133,198,076,049,234,-1,29028
```

UN AUTO-CAD PER DISEGNI ELETTRONICI

Un (lungo) programma che, sfruttando fino in fondo le risorse grafiche del C/128, può rappresentare un efficace strumento per disegnare agevolmente con il computer

di **Fabio Cavallin**

Chi possiede un C/128 avrà senz'altro notato il gran numero di comandi dedicati alla grafica ed alla gestione degli sprite e si sarà chiesto come utilizzarli al di fuori dei videogame.

Il programma di cui parliamo, interamente scritto in basic, permette di tracciare e salvare su drive (indispensabile) qualunque tipo di circuito elettrico o elettronico. Si può quindi realizzare un archivio molto utile sia per hobbysti che per studenti.

Mediante l'uso di un joystick in porta 1 è possibile muovere sullo schermo il simbolo prescelto, collocarlo dove si ritiene più opportuno e quindi "depositarlo" con la semplice pressione del pulsante Fire. In memoria sono collocati una trentina di simboli con la possibilità di ampliare il numero, tracciandone altri di qualsivoglia natura. Ma andiamo con ordine.

Dopo aver caricato il programma (dal dischetto "Directory") ed im-

partito il Run, si attenderà qualche secondo prima che compaia la schermata con il menu principale che offre all'utente ben 11 opzioni.

SOLO SU DISCO

Il lunghissimo programma inviato dal nostro lettore (a proposito, hai dimenticato di inviare il tuo indirizzo: comunicacelo!) rappresenta sicuramente un valido esempio di costanza ed impegno.

Purtroppo, come abbiamo già detto, l'eccessiva lunghezza del listato ne avrebbe scoraggiato la digitazione, per non parlare dell'operazione di debug.

E' stata quindi presa la decisione, ormai consolidata da un po' di tempo a questa parte, di inserire, sul dischetto "Directory" di questo mese, il listato ed i sette disegni realizzati dal nostro lettore.

Si noti che, per l'hard-copy, è stata usata una nostra routine l.m. (più volte pubblicata e menzionata su queste stesse pagine) allocata da 5392. Come intuitivo, non vi sono problemi di incompatibilità.

Il programma, di certo, non può esser considerato uno strumento "professionale" a causa dei limiti intrinseci del computer e della lentezza del Basic.

Tuttavia ne consigliamo lo studio (e l'utilizzo) a chi intende approfondire le proprie conoscenze sui comandi grafici del potente computer.

LE OPZIONI

1 DISEGNA

E' il comando principale mediante il quale si seleziona il numero del simbolo prescelto e lo si posiziona opportunamente sul video muovendolo con il joystick e premendo il pulsante. La pressione del tasto zero (0) permette di connettere i vari simboli tra loro tracciando una riga tra due punti individuati dal centro del mirino, che compare sul video, anch'esso movibile mediante joystick.

Dei due punti, il primo viene tracciato subito, in modo da avere un punto di riferimento quando si tracciano linee piuttosto lunghe. In ogni caso, quando compare la lettera "s" in alto a sinistra, bisogna dare conferma premendo, appunto, il tasto "s". Per ritornare al menu principale basta inserire un numero minore di zero.

2 VELOCITA'

Questa opzione informa l'utente sulla velocità di spostamento dei simboli (e del mirino) e fornisce la possibilità di variarla opportunamente.

3 VISUALIZZA

Questa scelta permette di esaminare con comodo la pagina grafica.

4 ELENCO SIMBOLI

Fornisce l'elenco dei simboli con i rispettivi numeri di riferimento.

5 SALVA DISEGNO

Al termine del tracciamento del circuito è necessario salvare la pagina grafica su disco. E' possibile anche salvare un circuito modificato assegnando lo stesso nome che possedeva prima della modifica. La lunghezza dei nomi è limitata a 12 caratteri a causa del prefisso "im. " (assegnato automaticamente dal programma) necessario al computer per riconoscere i circuiti salvati da altri programmi presenti sul disco.

6 CARICA DISEGNO

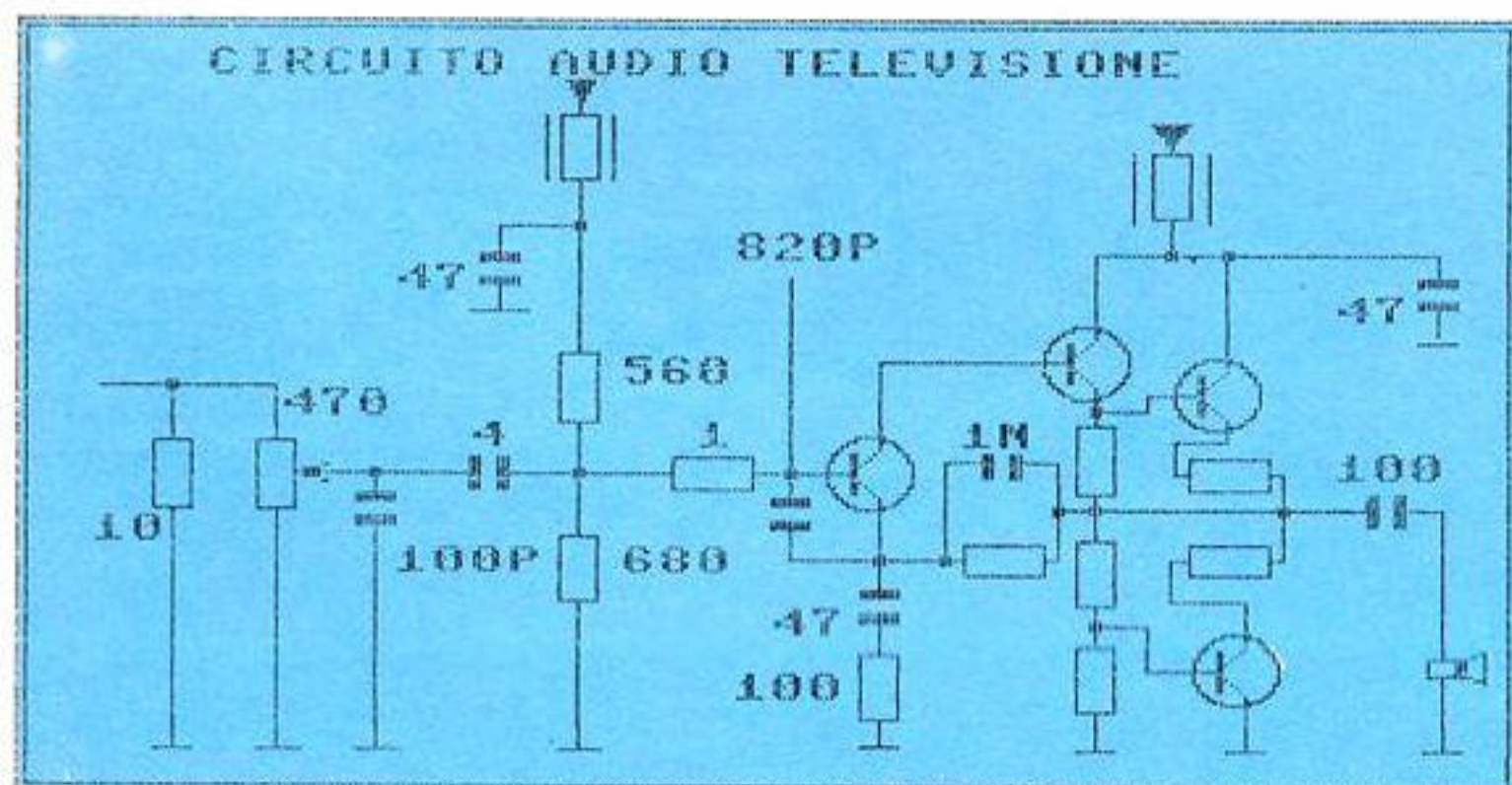
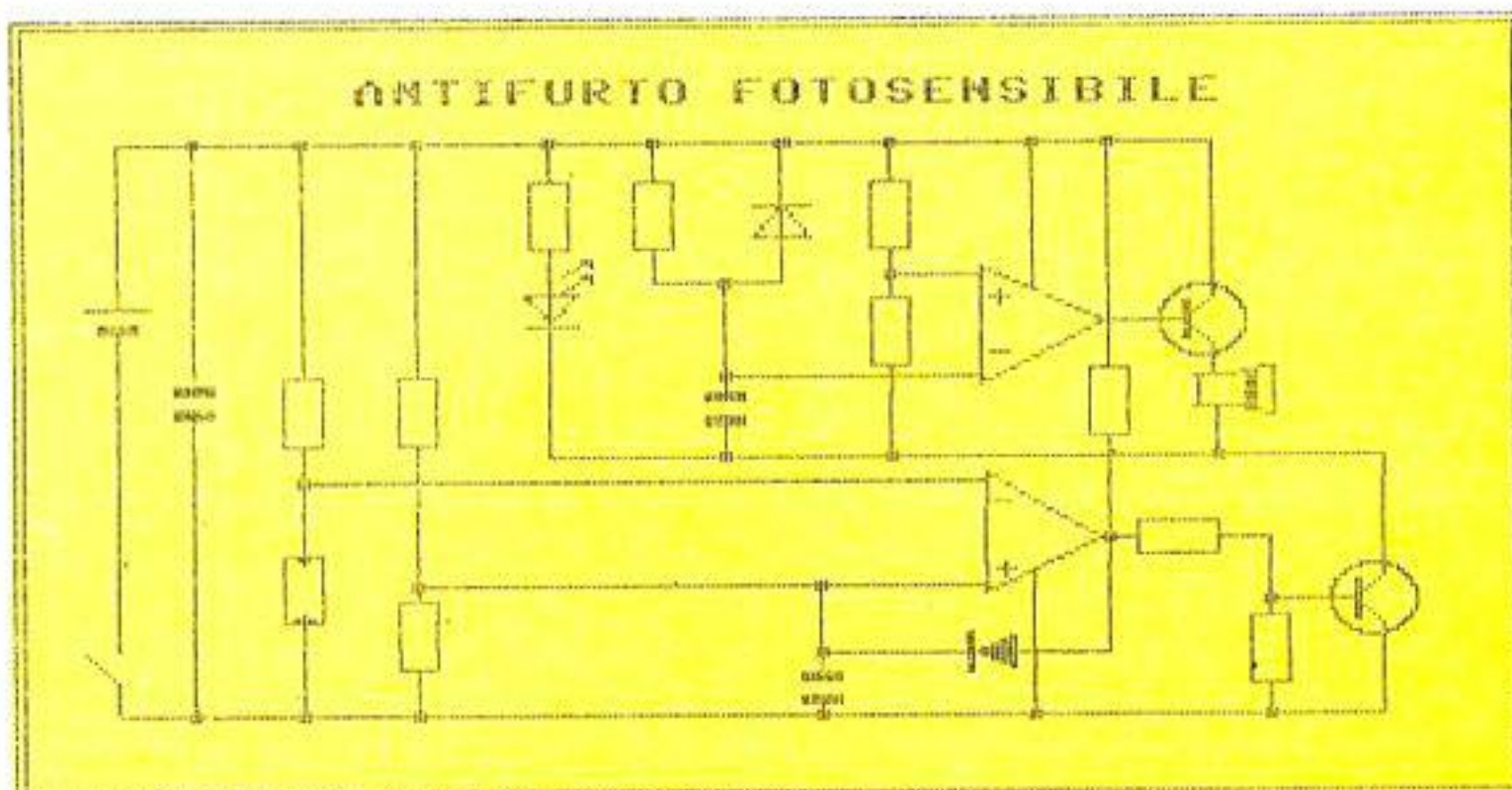
Per esaminare e/o modificare un disegno è necessario caricarlo in memoria battendo dapprima il nome PRIVO del prefisso "im. ". Nel caso in cui il computer non trovasse il file desiderato, ricompare la schermata con il menu principale.

7 CANCELLA DISEGNO

Viene cancellata completamente la pagina grafica, ma è richiesta, per sicurezza, una conferma.

8 INSERISCE TESTO

Permette l'inserimento di stringhe alfanumeriche che hanno il compito di completare i simboli con numeri, lettere o messaggi idonei a distinguere i componenti del circuito tracciato. Il cursore che appare è movi-

bile mediante il joystick e rappresenta la prima lettera della stringa inserita. Per ritornare al menu basta premere il tasto Return senza battere alcun carattere alla successiva richiesta.

9 CANCELLA PORZIONE
DI SCHERMO

Permette di cancellare il contenuto di un rettangolo di cui vanno indicate le coordinate degli estremi "alto a sinistra" e "basso a destra", da assegnare mediante il movimento del joystick. Anche in questo caso il primo punto viene tracciato subito, in modo da funzionare come riferimento.

"A" LISTA DISEGNI

Fornisce la lista dei disegni presenti su disco contenenti il prefisso specifico e, quindi, caricabili in memoria.

"B" FINE LAVORO

Si esce dal programma resettando, o meno, il computer.

COMMENTI

Come si può notare, è possibile gestire piuttosto agilmente il disegno, soprattutto dopo aver preso confidenza con i vari comandi.

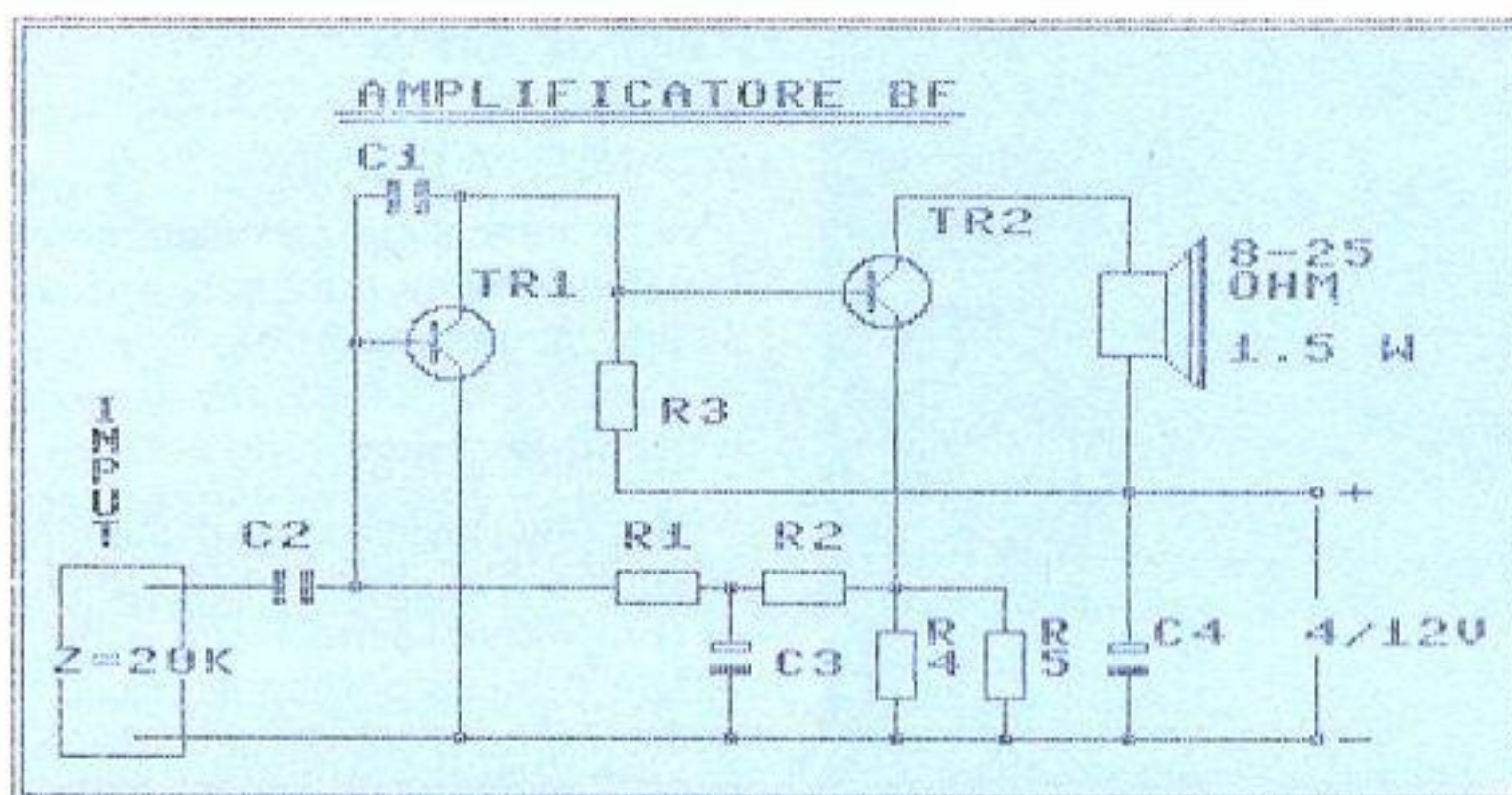
Il numero di simboli è facilmente ampliabile utilizzando le righe di programma da 248 a 260, iniziando con un gosub 300 e quindi disegnando il simbolo in una griglia di 24x21 pixel in alto a sinistra; si provvederà a memorizzare il tutto in una stringa A$(N), in cui "n" è il numero del simbolo, mediante un comando Sshape a$(n), 0, 0, 23, 20.

Occorre inoltre aggiornare la lista dei simboli (da riga 650 in poi) e ricordare che il programma prevede un massimo di 100 simboli (per valori maggiori bisogna dimensionare opportunamente la stringa A$ in riga 20).

Se per caso si dovesse uscire dal programma, resettando il computer senza aver salvato il disegno, niente paura: basta battere, in modo diretto...

```
bsave "im. nomedisegno", p8152
to p16384
```

...per non perdere il tempo speso sulla tastiera.

E' utile notare che premendo il tasto Run / Stop il programma non si interrompe ma riporta al menu principale. Questo particolare è utile soprattutto nei casi in cui sia stata selezionata un'opzione non voluta: premendo il tasto si annulla la scelta errata.

AMPLIFICATORE BF

C1 TR1 TR2 8-25 OHM 1.5 W R3 IMPUT C2 R1 R2 Z=20K C3 R4 R5 C4 4/12V

I limiti del programma sono essenzialmente due: la lunghezza del listato, l'impossibilità di ruotare i simboli disegnati e la conseguente necessità di doverli ridefinire per le varie direzioni desiderate.

Il listato, tuttavia, può rivelarsi senz'altro un valido aiuto per chi, per hobby o per studio, debba crearsi un archivio di circuiti elettrici.

NEGOZIO AL PUBBLICO
E VENDITA PER CORRISPONDENZA

CASH & CARRY

COMMODORE POINT '88

COMMODORE AMIGA 500/2000

"AL PREZZO PIÙ BASSO D'ITALIA" »CON GARANZIA & OMAGGIO«

Amiga VID a colori direttamente — L. 99.000
Digitalizzatore di immagini per Amiga 500/1000/200

Amiga Syntetic — L. 99.000
Digitalizzatore Audio per Amiga 500/1000/2000

Amiga VIDEOSOUND — L. 230.000
Digitalizzatore Audio VIDEO, tutto in uno.
Ottimo, per 500/1000

INT MIDI AMIGA PLUS — L. 65.000
Nuova interfaccia midi per Amiga 500 1000/2000

DRIVE AGGIUNTIVO AMIGA 500/1000 (SLIM LINE
L. 199.000

ESPANSIONE 512 PER AMIGA 500
INT. con orologio — (Telefonare)

ESPANSIONE 2MB per A500/1000
L. 850.000

EMULATORE 64 per AMIGA — L. 20.000
L'UNICO EMULATORE VERAMENTE FUNZIONANTE CON L'AUDIO E CON LA POSSIBILITÀ DI SALVARE I FILES SU 3 1/2

VID-AMIGA II — L. 899.000
Novità digitalizzatore in tempo reale per A 500/1000/2000

NOVITÀ **PENNA OTTICA PER AMIGA 500/1000/2000** — L. 149.000

NOVITÀ **DRIVE 5 1/4 per amiga** — L. 239.000
Permette di CARICARE pvg. del PC-IBM

DRIVE INTERNO PER AMIGA 2000 (MECC. NEC)
L. 179.000

NOVITÀ **VIDEON (AMIGA)** — L. 290.000
Nuovo digitalizzatore di immagini - Lavora senza filtri e in "PIÙ direttamente da Videoregistratore o Telecamera!

OLTRE 1000 PROGRAMMI AMIGA
RICHIEDERE CATALOGO

COMMODORE 64/128 - COMMODORE 64/128

THE NEW FINAL TURBO III per 64/128 (modo 64) L. 69.000
L'emulazione continua!!!
Eccovi l'ultima release della mitica cartuccia notevolmente migliorata e modificata. Turbola favolosa routine dello speddos su cartuccia fino a 10 volte più veloce sia in lettura che in scrittura!!! 8 tasti funzione programmati. 24 K ram per i grog. in Basic. Un favoloso protettore di programmi tipo O.M.A. incorporati. Dischi e cassette IN UN SOLO FILE!!! (boot se necessita. Inoltre ha incorporato il GAME KILLER (evita la collisione degli sprite ed ha ben 40 comandi Basic Turbo a disposizione ... HARDCOPY "HP". Premendo un solo tasto potrete fare l'ardcopy del video in 12 gradazioni di grigio.
ECCEZIONALE!!!

EPRON NEW GRAPHIC MPS 803
Si sotituisce il generatore di caratteri della stampante MPS-80 (per migliorare la leggibilità della scrittura con quattro nuovi set di caratteri.
L. 35.000

PROCESSORE VOCALE
L. 115.000
Digitalizzatore vocale tipo "Voice Master" notevolmente migliorato composto a cartuccia hardware e microfono software interamente in italiano con ampio manuale di istruzioni. Incredibile, fa parlare il tuo Commodore 64 puoi programmarlo a fin che conosca la tua voce e ti risponda.

KIT PULIZIA DRIVE
5 1/4 o 3 1/2
L. 15.000 cad.

KIKSTART 1.3
Rom Commodore
per tutti gli Amiga
L. 270.000

Programmatore Eprom per Amiga
L. 299.000

GENLOCK professionale
Broacasting novità assoluta per TV-Studi
L. 2990.000

Alimentatori professionali
----
per 64 — L. 39.000
per 128 — L. 69.000

VIDEODIGITAL 64
Nuovo digitalizzatore in cartuccia, digitalizza le tue più belle immagini con l'aiuto di una videocamera o videoregistratore semplicissimo da usare con manuale in italiano. Inoltre è possibile modificare le immagini con il KOALA ecc.
L. 70.000

STARDOS NEW! Eccezionale novità un velocizzatore che supera persino la velocità dello speed-dos attiva i tasti funzione ecc. In una sola Eprom kit da inserirsi nel C64 con manuale in ital. Non necessita di elaborazioni al drive ne del cavo parallelo.
L. 39.000

Adattatore telematico 64 Commodore V21-V23
TUTTO IN ITALIANO — L. 99.000

REALTIME DIGITAL 64 — L. 170.000
Notevolmente migliorato con possibilità di animazioni e videoclip.
IL PRIMO DIGITALIZZATORE
IN TEMPO REALE PER IL TUO 64/128

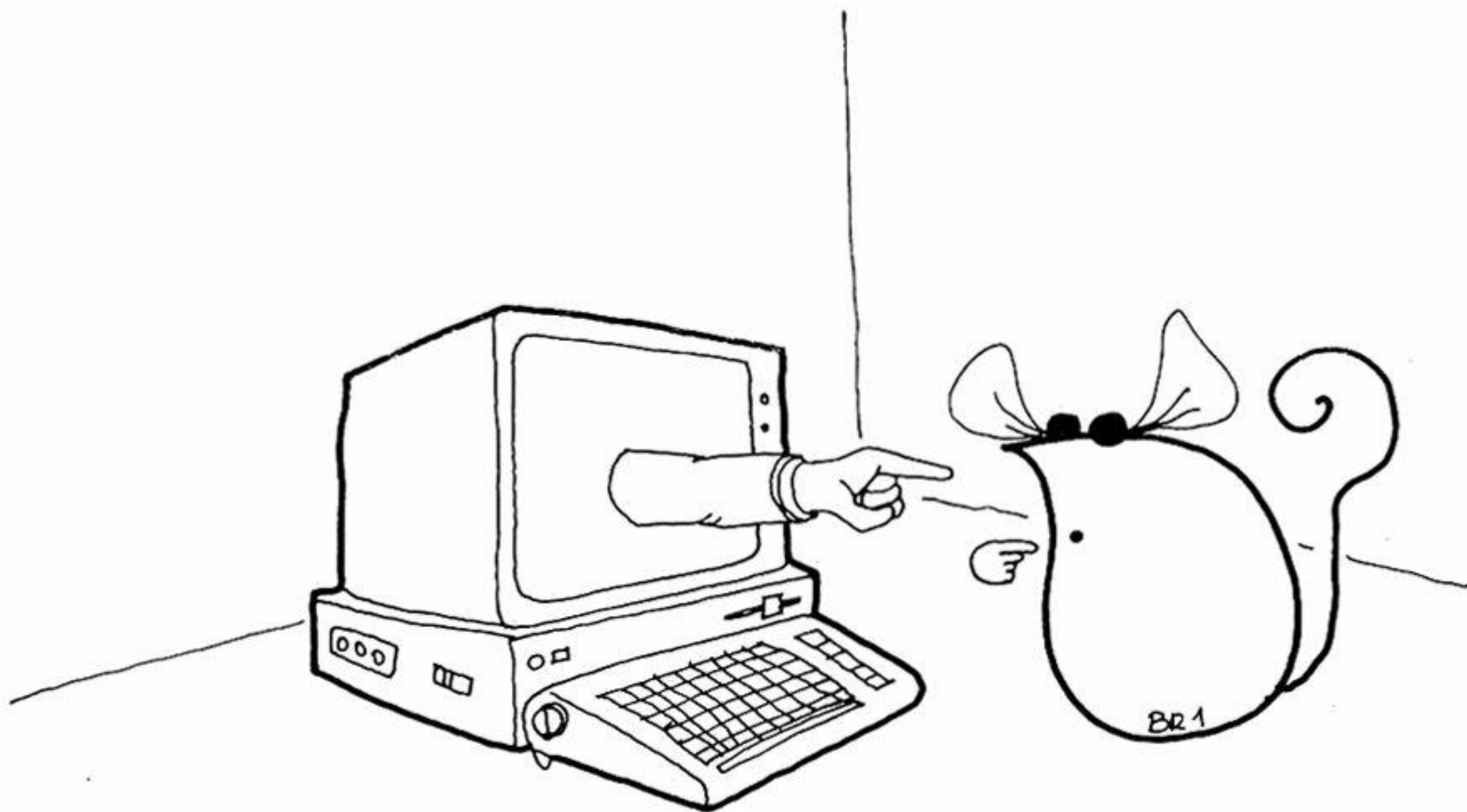
DRIVE OC-118 per 64/128
Drive Slim-line 100% compatibile
A PREZZO IMBATTIBILE

NOVITÀ **- MIKIE II -**
The CARTRIDGE (NOVITA) — L. 60.000
Nuova cartuccia multiutility con Turbo, Superturbo fino a 10 volte più veloce, Sprite Killer, Poker, Monitor Utility e comandi aggiuntivi. Toolkit, tasti funzione, copyfiles, ecc.

PIÙ UN ECCEZIONALE SPROTETTORE CASSETTA – DISCO E VICEVERSA IL TUTTO IN UN UNICO FILE
TUTTO IN ITALIANO !!!
Compatibile speed-dos e drive compatibili

NOVITÀ **NOVITÀ ASSOLUTA!!!**
The ICONE CARTRIDGE
per 64/128
L. 73.000
Un nuovo sistema operativo a ICON e finestre grafiche che racchiudono le caratteristiche della final III & The cartridge 64 e molto più!!
Potrete valorare quasi come con il Geos 64
(MAN. IN ITALIANO)

IN OFFERTA

O.M.A. PLUS	L. 60.000
TURBO FREEZE	L. 50.000
SPEED-DOS 1541	L. 49.000
SPEED-DOS 1541/C - OC 118	L. 59.000
PENNA OTTICA	L. 49.000
TAST LOAD - RESET	L. 30.000
KIT EPROM 80	L. 25.000
KIT GRAFICO 802	L. 35.000
RESET 64	L. 10.000
COPRITASTIERA 64	L. 10.000
COPRITASTIERA 64 NEW	L. 15.000
COPRITASTIERA 128	L. 18.000

METTERSI IN EVIDENZA

Sfruttiamo gli sprite per un'utile e didattica applicazione

di **Michele Maggi**

Per tutti i neofiti del computer il cruccio maggiore è senz'altro rappresentato dai programmi che contengono numerose linee DATA. Decine e decine di righe spesso piene di numeri che, senza significato apparente, appesantiscono i listati che il povero principiante si appresta a digitare.

Dopo la digitazione, se non bastasse, al primo RUN è tutto un susseguirsi di "Out of data" e "Illegal quantity".

Chiunque abbia avuto il sadico piacere di leggere questi messaggi di errore sa, per esperienza, quanta fatica e impegno richieda la fase di rilettura e controllo del listato, fase resa ancor più difficile dal dover fissare lo sguardo ora sul monitor, ora sulla pubblicazione dalla quale è tratto il programma.

L'inconveniente più comune è quello definito "salto della riga" che consiste nel saltare involontariamente (con gli occhi) da una riga all'altra con il conseguente rischio di digitare la prima metà di una linea e la seconda metà della successiva.

A questo inconveniente si pone facilmente rimedio utilizzando un righello da far scorrere sulle righe della rivista in modo da non perdere il "segno".

Purtroppo in questo modo si risolve solo metà del problema in quanto, sul monitor, è ben difficile utilizzare alla stessa maniera un righello.

Un piccolo aiuto ci viene proprio daglisprite che, se opportunamente manipolati, possono trasformarsi in una specie di "evidenziatore", con la funzione di mettere in risalto la riga in cui si trova il cursore.

Il funzionamento della routine principale è, ovviamente, in interrupt; agisce in modo da convertire la posizione Y del cursore, misurata in numero di righe di schermo (contenuto nella locazione 214, $D6), nel corrispondente formato "pixel", necessario per gli sprite. Ciò si realizza grazie ai tre codici macchina "ASL" (che corrisponde alla moltiplicazione per 8) e aggiungendo, successivamente, $22; inutile dire che per tutti gli sprite la posizione X resta costante.

Per attivare (o disattivare) la routine basta digitare SYS 49152.

Il disassemblato è sufficientemente commentato e se ne consiglia lo

SCHEDA TECNICA

Software di utilità generale

Hardware richiesto: C/64. Non adattabile ad altri computer Commodore

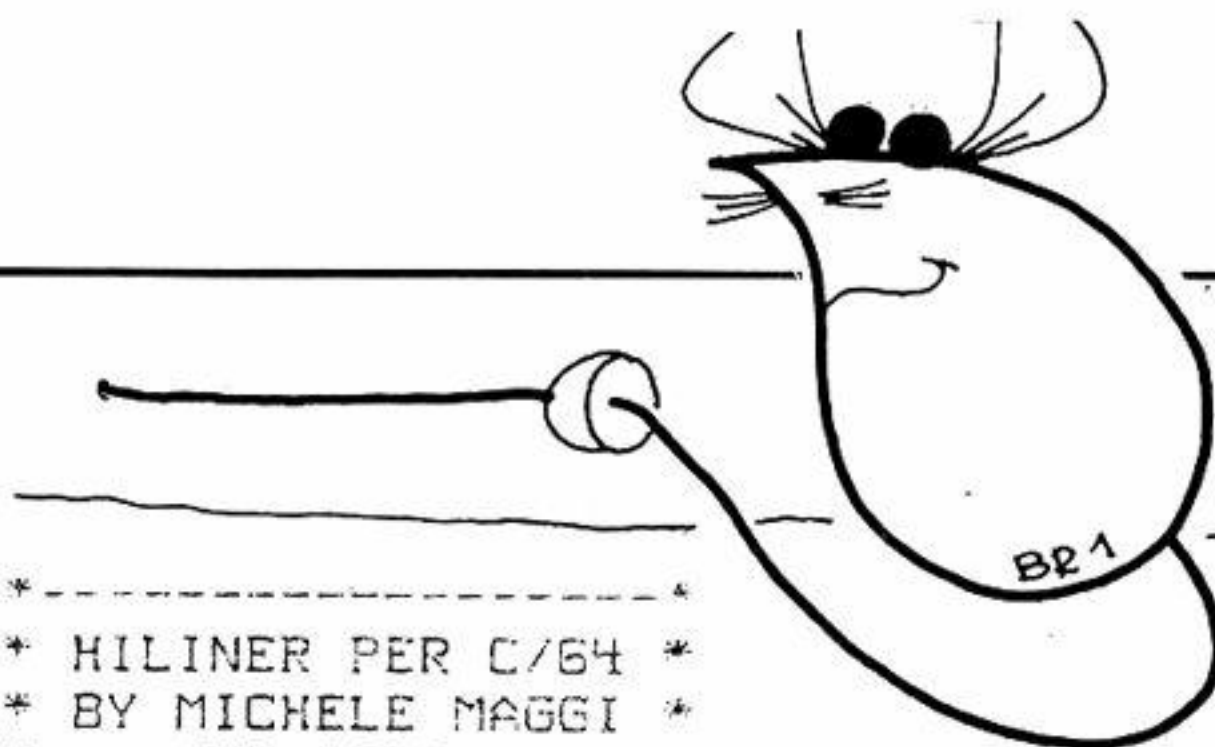
Consigliato a coloro che vogliono approfondire la tecnica di interrupt

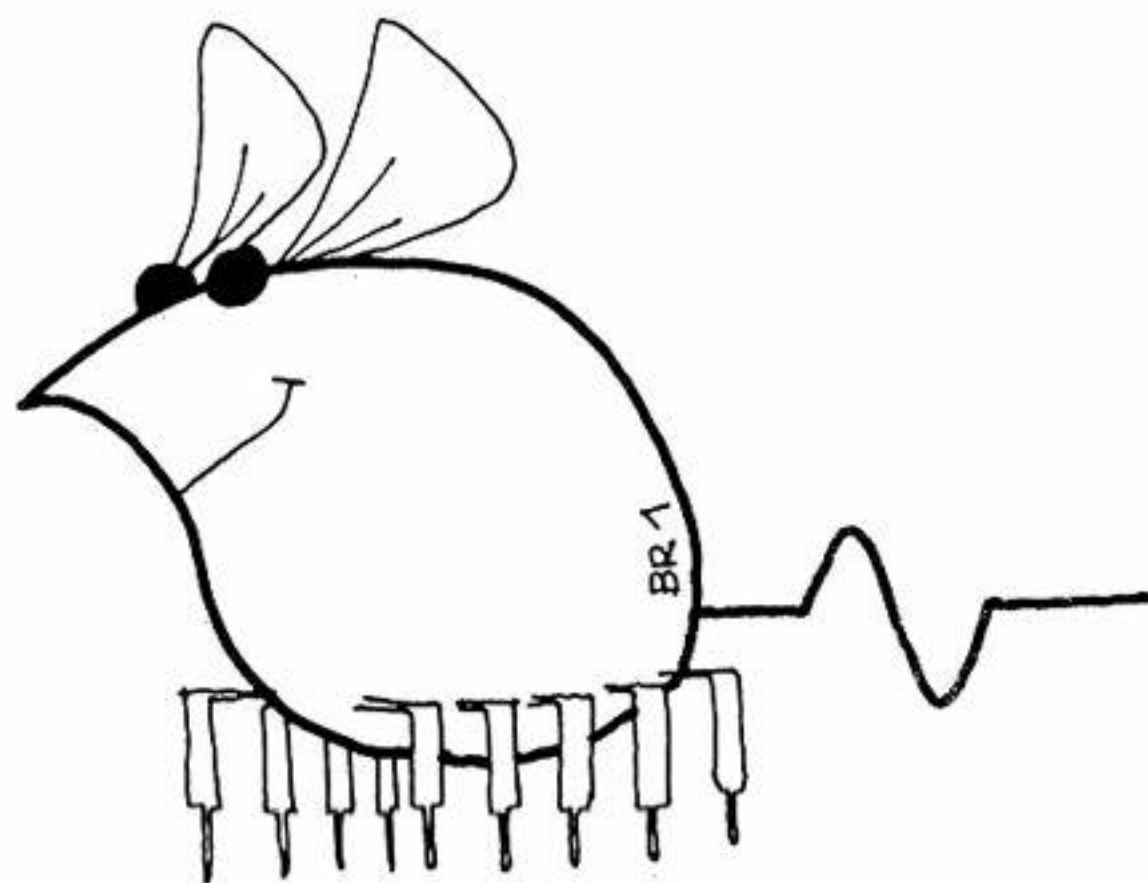
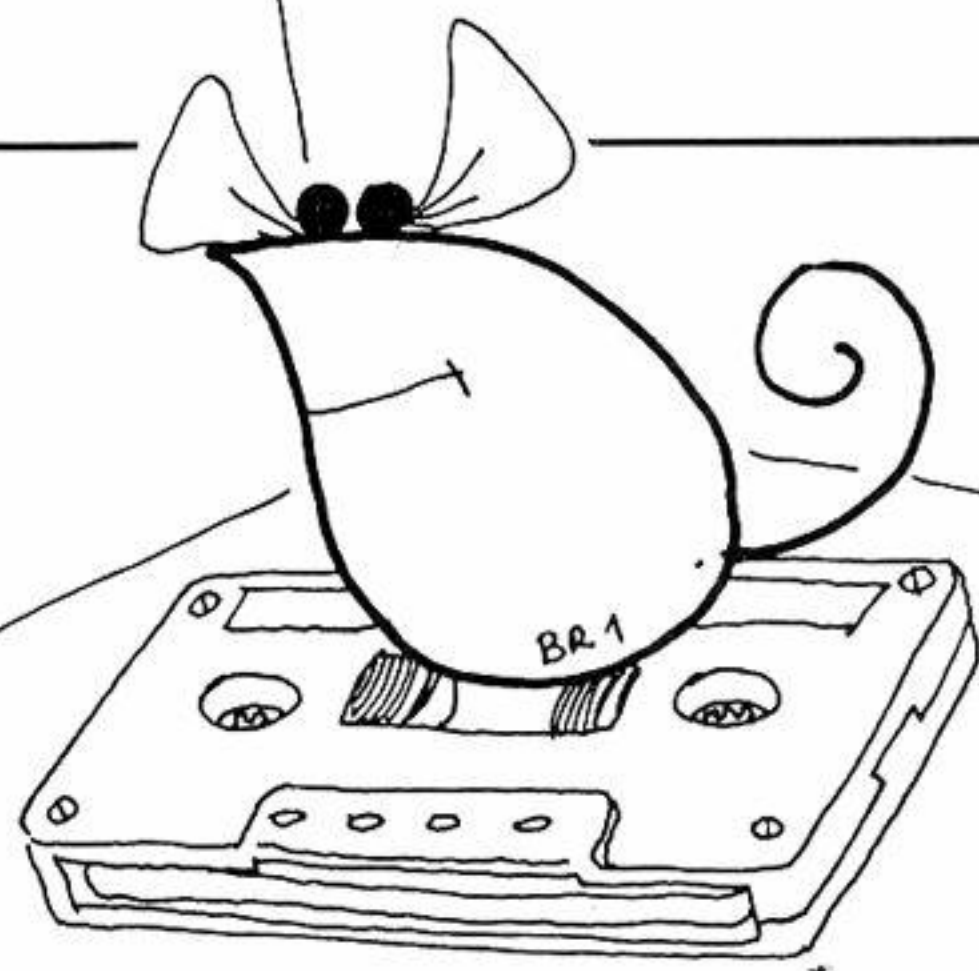
Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

studio anche a coloro che, pur non essendo interessati all'applicazione, intendono approfondire le proprie conoscenze riguardo all'interrupt.

UN PICCOLO INCONVENIENTE

A causa di un "bug" del Sistema Operativo le operazioni di Load / Save da disco vengono disturbate dalla presenza di più sprite allineati sullo schermo (vedi, a questo proposito, l'articolo "Lo strano caso di Doctor 64 e mr. Bug", C.C.C. n. 52); si consiglia, pertanto, di disattivare Hiliner (anche con un semplice Run / Stop e Restore) prima di compiere qualsiasi operazione di I/O su disco.

Gli sprite, ovviamente, possono essere manipolati come si vuole, magari colorando alcuni ed altri no, rendendone visibili alcuni e assegnando loro, o meno, la priorità di visualizzazione.

```
10 REM *--------------------*
20 REM * HILINER PER C/64 *
30 REM * BY MICHELE MAGGI *
40 REM *      (C) 1988      *
50 REM *--------------------*
60 :
100 FOR I=0 TO 160
110 READ A: CK=CK+A
120 POKE 49152+I,A
130 NEXT
135 :
140 IF CK=20364 THEN 160
150 PRINT"ERRORE NEI DATA!":END
155 :
160 PRINTCHR$(147)
165 PRINT"HILINER ATTIVATO!!!
170 PRINT"SYS 49152 PER ATTIVARE/DISATTIVARE
175 PRINT"53269  ACCENSIONE/SPEGNIMENTO"
176 PRINT"53275  PRIORITA' SCHERMO/SPRITE"
180 SYS 49152
190 END
200 :
1000 DATA 076,106,192,032,137,192,169
1001 DATA 011,160,007,153,247,007,136
1002 DATA 208,250,169,024,160,000,024
1003 DATA 153,000,208,105,048,200,200
1004 DATA 144,247,169,096,141,016,208
1005 DATA 169,008,141,010,208,169,056
1006 DATA 141,012,208,169,127,141,028
1007 DATA 208,141,027,208,141,029,208
1008 DATA 141,023,208,141,021,208,169
1009 DATA 001,141,037,208,120,169,080
1010 DATA 160,192,141,020,003,140,021
1011 DATA 003,088,096,024,173,134,002
1012 DATA 141,038,208,165,214,010,010
1013 DATA 010,105,034,160,014,153,255
1014 DATA 207,136,136,208,249,076,049
1015 DATA 234,173,160,192,073,255,141
1016 DATA 160,192,173,160,192,240,140
1017 DATA 120,169,049,141,020,003,169
1018 DATA 234,141,021,003,088,169,000
1019 DATA 141,021,208,096,162,000,138
1020 DATA 157,192,002,232,224,063,208
1021 DATA 248,170,169,085,157,216,002
1022 DATA 232,224,012,208,248,096,255
1023 END
```

```
1  *----------------------------------*
2  * HILINER PER C/64-128             *
3  * EVIDIENZIA LA RIGA CORRENTE      *
4  * DEL CURSORE                      *
5  * BY MICHELE MAGGI                 *
6  * (C) 1988 SYSTEMS                 *
7  *----------------------------------*
8
9             JMP   MAIN       ;
10 START                       ;
11            JSR   SETSPRITE  ;
12            LDA   #$0B       ;SETTA I
13            LDY   #$07       ;PUNTATORI
14 LOOP01                      ;AGLI SPRITE
15            STA   $07F7,Y    ;AL VALORE 11
16            DEY              ;
17            BNE   LOOP01     ;
18            LDA   #$18       ;POSIZIONA
19            LDY   #$00       ;I SETTE
20            CLC              ;SPRITE
21 LOOP02                      ;IN X
22            STA   $D000,Y    ;TENENDO CONTO
23            ADC   #$30       ;DELL'MSB
24            INY              ;PER GLI
25            INY              ;ULTIMI DUE
26            BCC   LOOP02     ;
27            LDA   #$60       ;
28            STA   $D010      ;
29            LDA   #$08       ;
30            STA   $D00A      ;
31            LDA   #$38       ;
32            STA   $D00C      ;
33            LDA   #$7F       ;SETTA
34            STA   $D01C      ;MULTICOLOR
35            STA   $D01B      ;PRIORITA'
36            STA   $D01D      ;ESPANSIONE IN X
37            STA   $D017      ;ESPANSIONE N Y
38            STA   $D015      ;ACCENDE SPRITE
39            LDA   #$01       ;SETTA IL .
40            STA   $D025      ;MULTICOLOR 1
41            SEI              ;DIROTTA
42            LDA   #<ENTRY    ;I VETTORI
43            LDY   #>ENTRY    ;DELL'IRQ
44            STA   CINV       ;ALLA NUOVA
45            STY   CINV+1     ;ROUTINE
46            CLI              ;
47            RTS              ;
48 ENTRY                       ;
49            CLC              ;
50            LDA   $0286      ;SETTA IL
51            STA   $D025      ;MULTICOLOR 2
52            LDA   $D6        ;E POSIZIONA
53            ASL              ;GLI SPRITE
54            ASL              ;ALLA POSIZIONE
55            ASL              ;DEL CURSORE
56            ADC   #$22       ;
57            LDY   #$0E       ;
58 LOOP03                      ;
59            STA   $CFFF,Y    ;
60            DEY              ;
61            DEY              ;
62            BNE   LOOP03     ;
63            JMP   IRQ        ;ESCE A IRQ
64 MAIN                        ;
65            LDA   FLAG       ;ATTIVA LA
66            EOR   #$FF       ;ROUTINE
67            STA   FLAG       ;SOLO SE E'
68            LDA   FLAG       ;DISATTIVATA
69            BEQ   START      ;E VICEVERSA
70            SEI              ;RESETTA
71            LDA   #$31       ;I VETTORI
72            STA   CINV       ;DI INTERRUPT
73            LDA   #$EA       ;
74            STA   CINV+1     ;
75            CLI              ;
76            LDA   #$00       ;SPEGNE
77            STA   $D015      ;GLI SPRITE
78            RTS              ;ED ESCE
79 SETSPRITE                   ;CREA
80            LDX   #0         ;IL DISEGNO
81            TXA              ;DEGLI SPRITE
82 LOOP04                      ;
83            STA   704,X      ;
84            INX              ;
85            CPX   #63        ;
86            BNE   LOOP04     ;
87            TAX              ;
88            LDA   #85        ;
89 LOOP05                      ;
90            STA   728,X      ;
91            INX              ;
92            CPX   #12        ;
93            BNE   LOOP05     ;
94            RTS              ;
95 IRQ        =     $EA31
96 CINV       =     $0314
97 FLAG       DFB   255
```

Questo mese:

- Dark Castle
- Super Action Arcade Hits
- Mr. Hat
- Bionic commando
- Ransack
- Work Games
- Skate Crazy

Ogni mese, su queste pagine, verranno esaminati e testati i videogame più recenti per i computer Commodore 64 ed Amiga.

Ad ogni descrizione verranno associate una immagine, catturata tra le più belle schermate, ed una breve pagella.

Quest'ultima, pur se, inevitabilmente, frutto di impressioni personali di chi esamina il gioco stesso, ha lo scopo di assegnare una valutazione del livello del software,soprattutto tenendo conto di altri game analoghi disponibili sul mercato.

La pagella, comprende cinque voci:

IMPATTO: indica il livello di interesse suscitato dalla presentazione e dal tema del gioco.

SCENARIO: riguarda l'accuratezza con cui è realizzata la grafica e l'efficacia dei disegni degli sprite.

SUONO: valuta gli effetti sonori presenti e le eventuali musiche di sottofondo.

INTERESSE: si riferisce al livello di interesse che il gioco può suscitare in un giocatore abituato ai videogame e, implicitamente, alla sua probabilità di "permanenza" sui vostri monitor.

TOTALE: ha lo scopo di sintetizzare i precedenti valori con un voto unico.

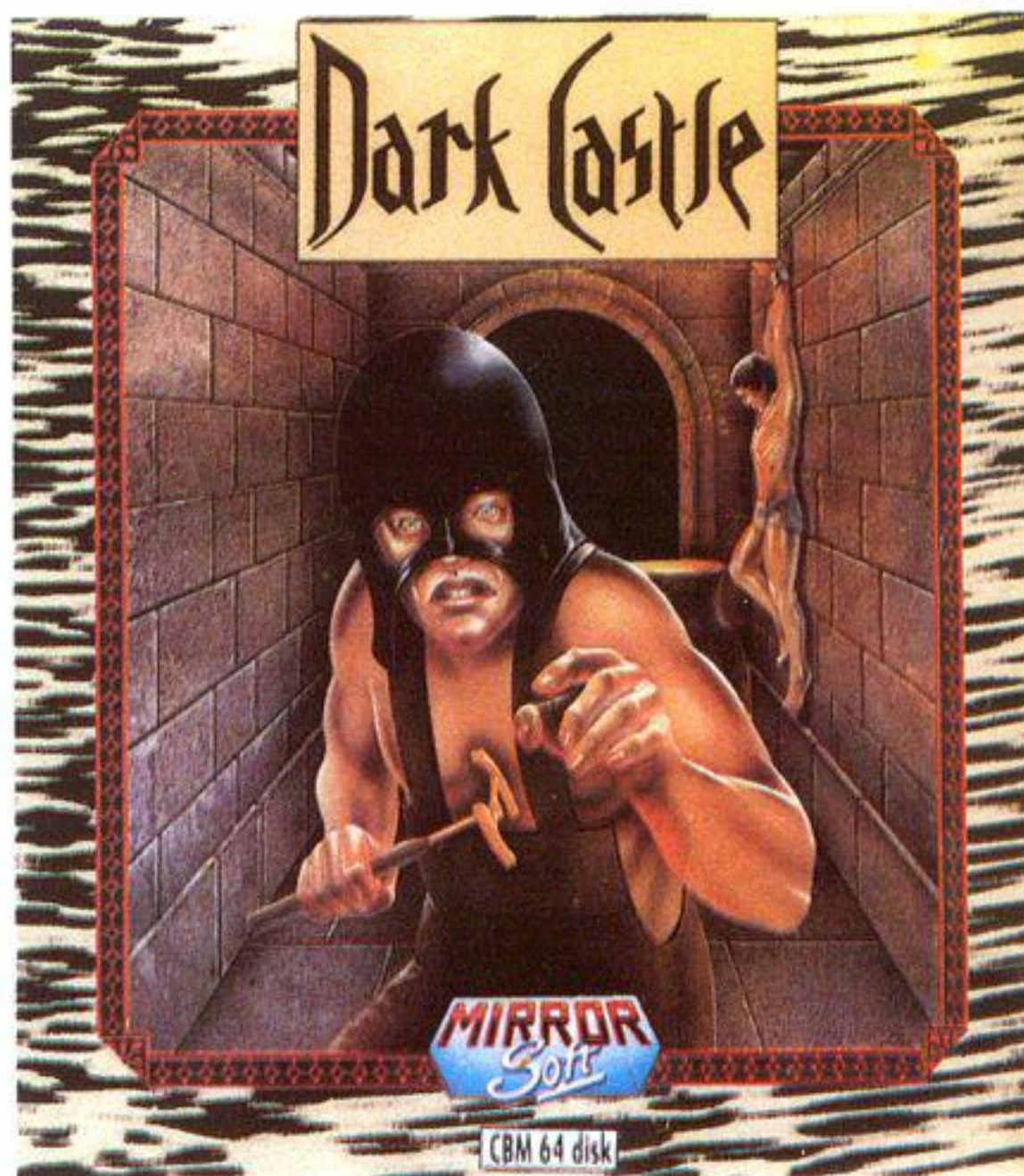
DARK CASTLE C-64/128

Sir Reginald, un cavaliere errante, dopo un dì un lungo giorno di fatiche, si appisola, in una foresta, al riparo dell'ombra di un albero.

Al suo risveglio si ritrova nel vasto salone di un misterioso castello in cui vi sono molte porte ma nessuno in grado di spiegargli che cosa sia successo e, soprattutto, come uscire da quel luogo.

Il compito del giocatore è quello di aiutare il nostro eroe nelle mille insidie e trabochetti che si celano tra le oscure mura del maniero.

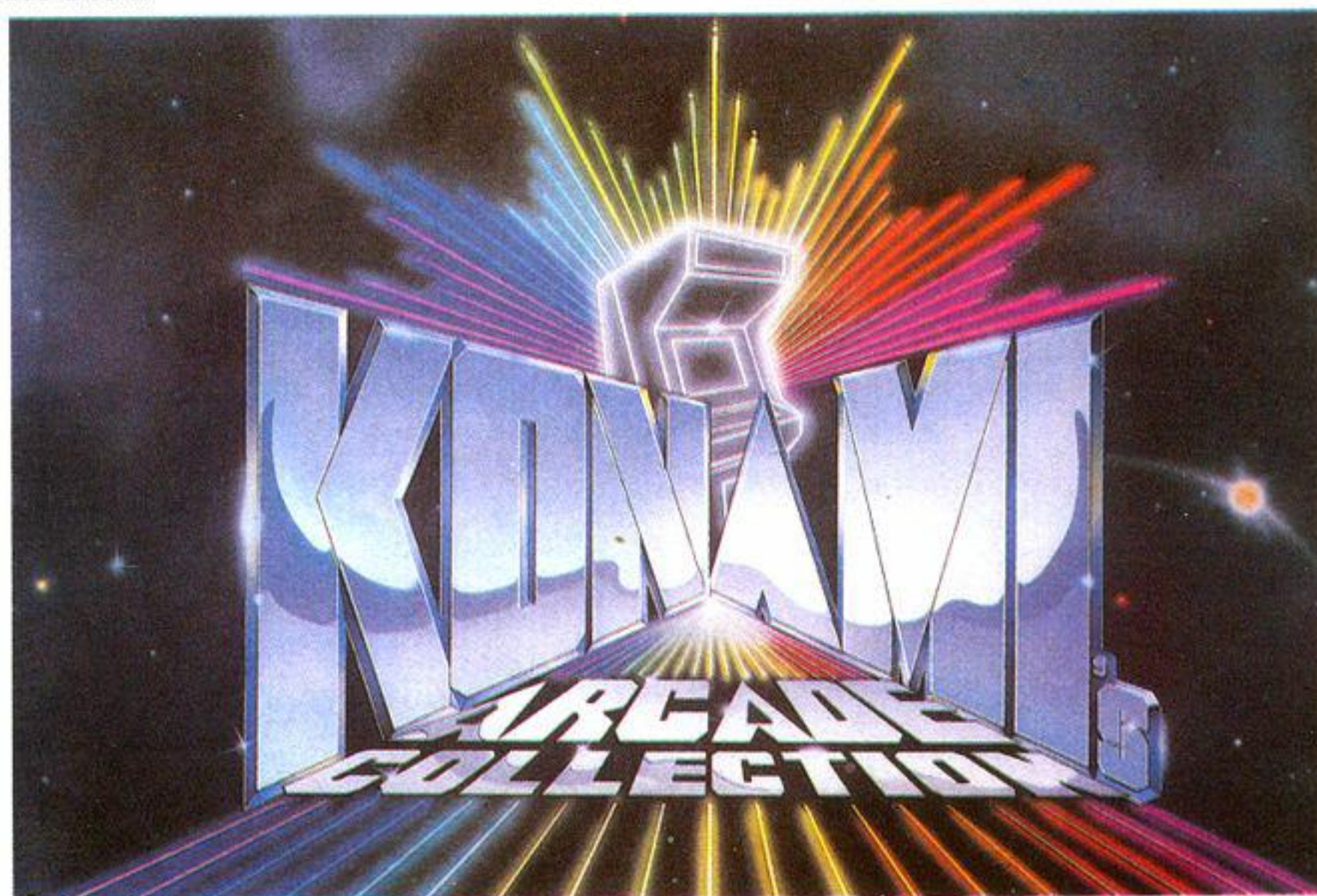
Ottimi gli effetti sonori mentre la grafica, in alcune schermate, è troppo minuta.

IMPATTO:	*7*
SCENARIO:	*6*
SUONO:	*8*
INTERESSE:	*7*
TOTALE:	**7**

SUPER ACTION ARCADE HITS C/64-128

E' una compilation dei migliori giochi prodotti dalla Konami, raccolti in una confezione di due dischi.

I titoli sono i seguenti:

- Jail Break
- Green Beret
- Yie ar kung fu
- Ping Pong
- Yie ar kung fu-II
- Nemesis
- Shao-lin's road
- Hypersports
- Mikie
- Jackall

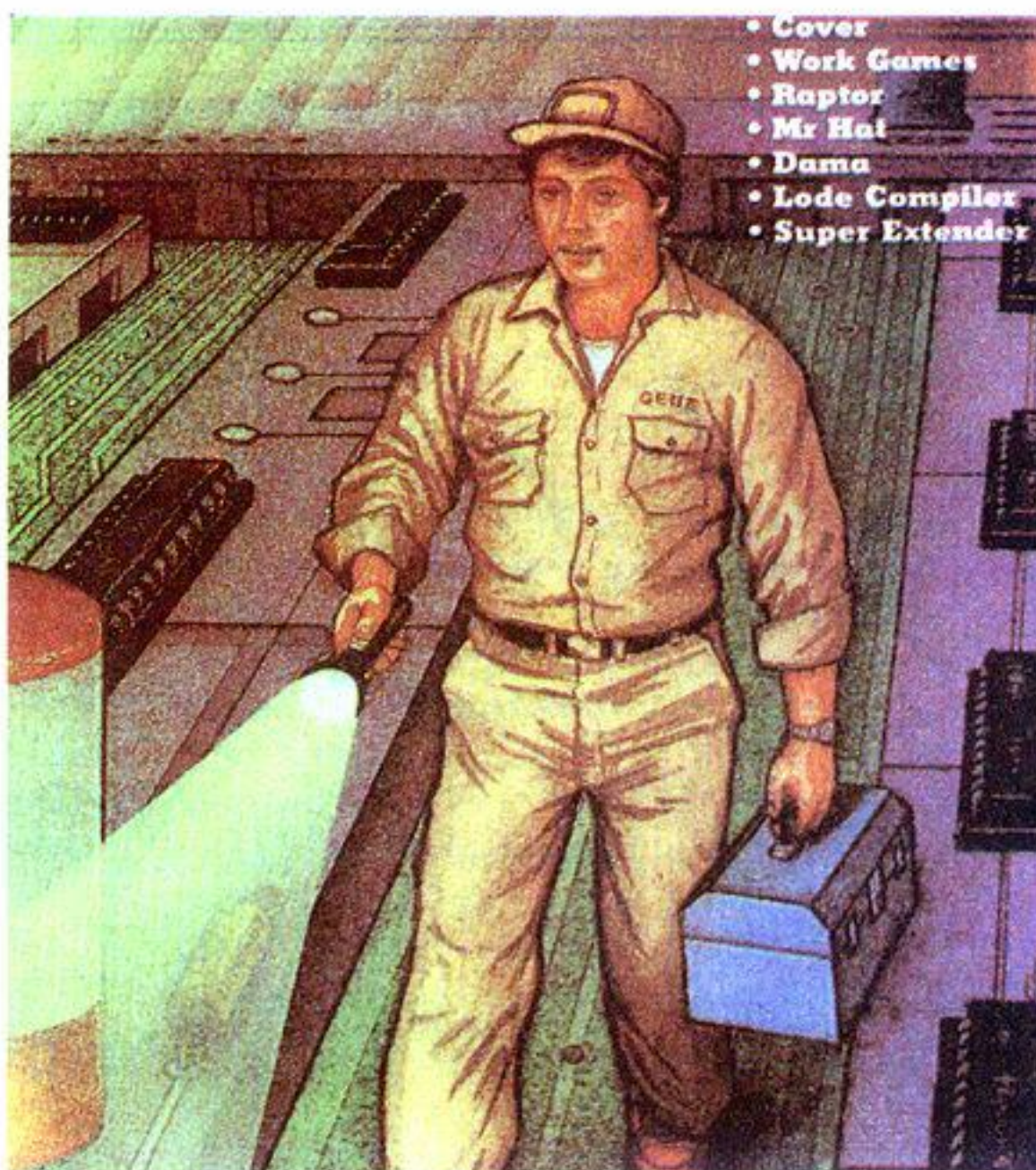
La collezione non può certo mancare a chi ama i giochi d'azione.

MR HAT
C-64/128

Un simpatico omino, con un grosso cappello che gli copre quasi per intero il corpo, ad esclusione delle gambe, è il protagonista di questo gioco.

L'obiettivo è riuscire ad impadronirsi del cappello d'oro, che si trova da qualche parte, all'interno di un insidioso labirinto.

Aggirandosi nei meandri, Mr Hat dovrà raccogliere alcuni oggetti, indispensabili per il proseguimento del gioco, prestando però attenzione ad altri che invece sono molto pericolosi.

Riuscirai in questa impresa?

Il gioco fa parte del disco "Commodore 64 Club n.6".

WORK GAMES
C/64-128

La ditta presso la quale lavori ha indetto una gara, aperta a tutti i dipendenti, per eleggere lo stakanovista dell'anno.

Tutti si sono messi in lista per le prove, che si terranno all'interno dell'azienda stessa, e che comunque fanno parte del normale ciclo produttivo.

Anche tu, come gli altri, ti iscrivi alla gara, dal momento che c'è in palio un grosso premio.

Per rendere più divertente la competizione, è stata prevista l'opzione per più giocatori, fino ad un massimo di 6.

Un videogame decisamente diverso dal solito!

BIONIC COMMANDO
C-64/128

A dieci anni dalla catastrofica guerra contro l'impero di Zargon, quando la Terra fu devastata da missili nucleari, ecco che appare uno spiraglio di luce, che illumina la speranza, dei pochi sopravvissuti, di ripopolare il pianeta.

Questa speranza è data dai "Bionic Commando", truppe di soldati bionici sofisticatissimi, dotati di numerosi sensori, radar ed armi laser.

Il futuro dell'umanità dipende ora dal successo della missione affidata ai droidi!

Disponibile su disco e cassetta.

IMPATTO:	*7*
SCENARIO:	*7*
SUONO:	*8*
INTERESSE:	*7*
TOTALE:	**7**

RANSACK
C-64/128

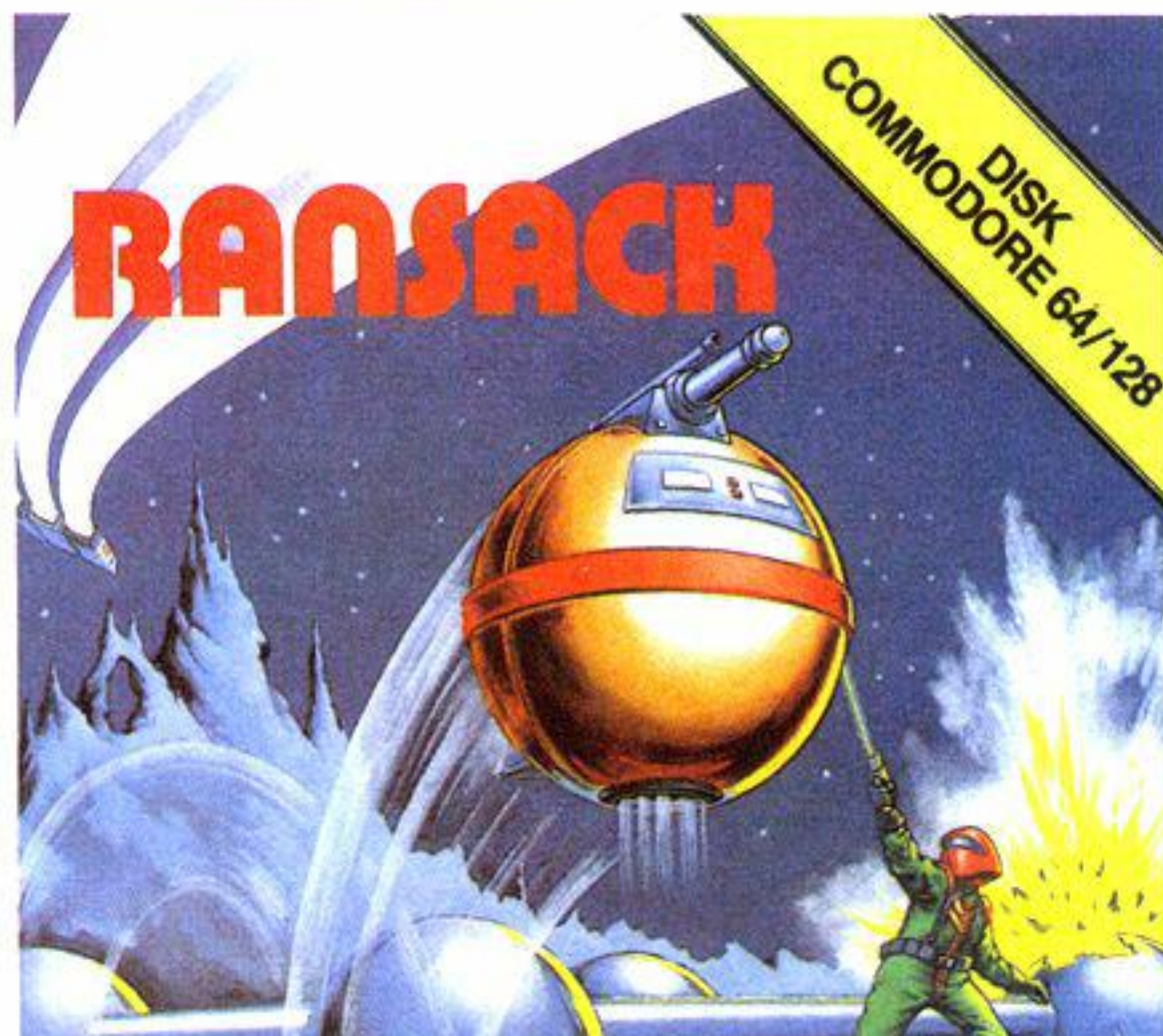
Il sistema solare Ryvian, da secoli in perfetta armonia, sta per trasformarsi in uno scenario di guerra, a causa di ribellioni in otto dei suoi pianeti.

Tu, al controllo di AL, un sistema da combattimento computerizzato, sarai impegnato in una missione che prevede l'intervento, sugli otto pianeti in questione, per domare le rivolte.

AL, pur essendo ottimamente equipaggiato con armi moderne, è difficilmente manovrabile, ed è facile distrarsi a causa della difficoltà di guida!

Nel complesso è un gioco ben architettato, anche se l'idea non è molto originale; peccato però che sia difficilmente giocabile... potrebbe essere una sfida per i più audaci!

Diponibile su disco.

IMPATTO:	*7*
SCENARIO:	*7*
SUONO:	*8*
INTERESSE:	*5*
TOTALE:	**6**

SKATE CRAZY
C-64/128

Freddy, un vostro simpatico amico, decide di partecipare ad una dura gara di pattinaggio su rotelle, percorrendo un tracciato pieno di ostacoli di varia natura.

Una giuria, composta da quattro giudici, analizzerà ogni singola manovra eseguita da Freddy.

Dal momento che i giudici convocati alla gara, sono abituati a veder pattinare ragazzi molto in gamba, piccole prodezze o salti eseguiti senza particolari acrobazie, non li impressioneranno, impedendogli di incrementare il punteggio.

IMPATTO:	*7*
SCENARIO:	*7*
SUONO:	*6*
INTERESSE:	*6*
TOTALE:	**6**

NUOVO CATALOGO

Un nuovo catalogo, in cui trovano posto tutti i prodotti ed i servizi Commodore, è stato presentato in occasione del SIM e dello Smau. A differenza delle precedenti edizioni, in cui le varie notizie erano ripartite in numerose brochure separate, il voluminoso fascicolo a colori meglio si presta per offrire una panoramica completa del mondo Commodore. Dal C/64 al genlock per Amiga, dal più piccolo Ms-Dos compatibile al sistema 386, tutti i prodotti vengono presentati in modo esauriente, all'utente potenziale, che ha modo anche di prendere nota degli indirizzi più interessanti: Commodore Point, Commodoctors; oltre alle notizie sulla garanzia e al servizio telefonico Hot Line offerto gratuitamente agli utenti.

LA STAMPA PARLA

Una pubblicazione trimestrale, diretta esclusivamente agli operatori commerciali di informatica, è destinata a raccogliere e divulgare le numerosissime notizie sul pianeta Commodore, pubblicate dalla stampa nazionale. Il nuovo periodico è l'ennesima dimostrazione che la presenza Commodore è ormai una realtà concreta, con la quale sono costrette a misurarsi tutte le altre aziende del settore. Gli "estratti" dai più importanti quotidiani, settimanali e mensili occupano le quasi 80 pagine del nuovo periodico, gestito direttamente dalla Commodore Italia.

OLTRE I DUECENTO

Numerosissime sono le nuove adesioni all'iniziativa "Commodore Point" che vede protagonisti i più importanti centri di vendita di informatica sul territorio nazionale. Per far fronte alle nuove, numerose, richieste è stato potenziato il settore con l'assunzione di nuovo personale particolarmente esperto nel trattare il lato commerciale della pacifica "invasione" Commodore. Ciò rientra, come accennato qualche numero fa, nella politica di espansione ormai in atto da tempo.

QUASI PRIMO

Massimo Bonfante, a causa di una rovinosa caduta nell'ultima gara, è giunto secondo nella massacrante corsa delle 125 monocilindriche. Il punteggio, nonostante ciò, è di tutto rispetto: 48 punti (il primo in classifica ha totalizzato 51 punti). Al bravo Massimo le nostre congratulazioni e la speranza di rivederlo ancora, nella prossima stagione agonistica, con i colori rosso e blu della Commodore, che ha sponsorizzato la sua Honda 125.

DIFFIDARE DELLE IMITAZIONI

Sembra che alcuni negozi (ovviamente non autorizzati) vendano "imitazioni" di adattatori telematici 6499 per C/64 che, simili nell'aspetto esteriore al prodotto originale, presentano inconvenienti di vario tipo. Nell'acquistare un adattatore, pertanto, la Commodore Italia suggerisce di esaminare attentamente il 6499 e di verificare, soprattutto, la presenza della cedola di garanzia originale nella confezione.

QUANTO COSTA IL TUO COMMODORE

LISTINO PREZZI HOME COMPUTER

Codice	Prodotto	Prezzo (IVA esclusa)
	Personal Computer	
A 500	Personal Computer 16/32 BIT CPU 512K RAM incorpora un floppy disk drive da 3" 1/2 da 880Kb e un mouse	L. 950.000
A 501	Espansione di memoria per Amiga 500 512Kbytes e orologio	L. 212.500
A 520/1	*Modulatore per Amiga 500 permette di collegare Amiga 500 ad un qualsiasi televisore b/n e colori*	L. 40.000
A SCART	Cavo collegamento Amiga 500 con TV prescart	L. 27.000
Nuovo C64	Nuovo personal computer CPU 64K RAM computer ad alta risoluzione grafica, 256 combinazioni di colori, sintetizzatore di suono. Collegabile ad un qualsiasi televisore	L. 375.000
1764	Espansione di memoria per C64 256Kbytes	L. 195.000
C 128D	Personal computer CPU 128Kb RAM CPU 128Kbytes espandibile a 512Kbytes, 48Kbytes ROM, basic 7,0. Tastiera separata. Alta risoluzione grafica, 16 colori + 8 sprites. 40 80 (RGB) colonne. Programmabile in CP/M 3.0. Funzionante in modo C64. Floppy disk da 340Kb incorporato.	L. 895.000
1700	Espansione di memoria per C128. 128Kb	L. 165.000
1750	Espansione di memoria per C128. 512Kb	L. 235.000
	Registratori	
1530	Registratore compatibile C64 - C128 - C128D	L. 55.000
	Unità a dischi	
1541 II	Floppy disk drive 5" 1/4. Unità di memoria di massa, drive singolo, capacità 170Kbytes in linea. Compatibile con C64	L. 395.000
1581	Floppy disk drive 3" 1/2. Unità di memoria di massa, drive singolo da 3" 1/2, capacità 800Kbytes. Compatibile C64 - C128 - C128D	L. 420.000
A 1010	Floppy disk drive esterno 3" 1/2	L. 395.000
1802	*Monitor a colori, alta risoluzione, 14" con audio. Collegabile al C64, C128, C128D (40 colonne)*	
2080	Monitor a colori alta persistenza alta risoluzione, 14", con audio antiriflesso. Collegabile ad Amiga PC, C64, C128, C128D. 640x400 pixel, 4096 colori	L. 630.000
1084	*Monitor a colori. Alta risoluzione, 14", con audio, antiriflesso. Collegabile al C64 - C128 - C128D - Amiga - PC. 640x400 pixel, 4096 colori.*	*L. 545.000*
	Stampanti	
MPS 1200 S	Stampante. 80 colonne, 120 cps, bidirezionale, carta in modulo singolo e trascinamento modulo continuo per C64 - C128 - C128D	L. 480.000

MPS 1200 P	Stampante 80 colonne 120 cps, bidirezionale, carta in modulo singolo e trascinamento modulo continuo per A500	**L. 480.000**
MPS 1250	Stampante con stesse caratteristiche del modello MPS 1200 P, interfacce seriale RS232C e parallela Centronics di serie	**L. 495.000**
MPS 1500	Stampante a colori. 80 colonne, 130 cps, 4 colori, bidirezionale, carta in modulo singolo e trascinamento modulo continuo.	**L. 550.000**
	Accessori	
6499	Adattatore telematico omologato. Collegabile al C64, permette il collegamento a Videotel PGE-Banche Dati.	**L. 149.000**
1351	Mouse. Per C64, C128, C128D	**L. 99.000**
1311	Joystick. Comando per giochi	**L. 13.500**
1399	Joystick. 8 microswitch-autofire	**L. 29.000**

LISTINO PREZZI SISTEMI

Codice		Prezzo (IVA esclusa)
PCI	Microprocessore 8088 (coprocessore 8087 opzionale) frequenza clock 4,77MHz. RAM 512Kb. Espandibile a 640Kb; 1 floppy da 360Kb. Monitor a fosfori verdi 12".	**L. 945.000**
PC 10-III	Microprocessore Intel 8088 a 16 bit; clock 10 MHz; memoria RAM: 640 Kb; memoria ROM (BIOS): 8 Kb autoconfigurabile; sistema operativo MS-DOS 3.21; GW-Basic; scheda video integrata monocromatico/colore di serie con emulazioni CGA/MDA/Hercules via software; 2 floppy disk drive da 360 Kb; monitor monocromatico a fosfori verdi da 12"; porte seriale RS232C e paralella Centronics; tastiera avanzata 102 tasti con 12 tasti funzione; porta mouse, incluso controller per H.D.	**L. 1.490.000**
PC 10-IIIC	Stessa configurazione ma con monitor 14" a colori mod. 1084	**L. 1.890.000**
PC 20-III	Microprocessore Intel 8088 a 16 bit; memoria RAM da 640Kb; memoria ROM (BIOS) da 8Kb autoconfigurabile. Sistema operativo MS-DOS 3.21. Scheda video monocromatico/colore di serie con emulazioni CGA/MDA/Hercules via software. Un hard disk da 20Mb e un floppy disk drive da 360Kb. Monitor monocromatico a fosfori verdi da 12". Porta seriale RS232C parallela Centronics. Porta mouse incluso controller per H.D.; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 2.240.000**
PC 20-IIIC	Stessa configurazione ma con monitor 14" a colori mod. 1084	**L. 2.640.000**
PC 40/20 AT	Microprocessore 80286 a 16 bit; clock 10/6 MHz; memoria RAM dal 1 Mb; hard disk da 20 Mb; un floppy disk drive da 1,2 Mb; Sistema operativo MS-DOS 3.2, GW-Basic; scheda video monocromatico/colore 132 colonne AGA di serie; monitor monocromatico a fosfori verdi da 14"; porta seriale RS232C e parallela Centronics; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 3.490.000**
PC 40/40 AT	Microprocessore 80286 a 16 bit; clock 10/6 MHz; memoria RAM da 1Mb. Un hard disk da 40Mb; un floppy disk drive da 1,2Mb. Sistema operativo MS-DOS3.2, GW-Basic. Scheda video monocromatico/colore EGA WONDER di serie. Monitor monocromatico a fosfori verdi da 14". Porta seriale RS232 e parallela Centronics; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 4.490.000**
PC 60/40	Microprocessore 80386 a 32 bit; clock 8/16 MHz; memoria RAM da 2,5 MB; hard disk da 40 Mb; 1 floppy disk drive da 1,2 Mb; sistema operativo MS-DOS 3.20; scheda EGA WONDER di serie; monitor ADI monocromatico a fosfori verdi da 14"; 2 porte seriali RS232C e 2 parallele Centronics. In opzione: floppy disk drive da 3" 1/2 e 1.44 Mb. Tastiera avanzata 102 tasti con 12 tasti funzione	**L. 8.490.000**
PC 60/80	Stessa configurazione del PC 60/40 ma con hard disk da 80 Mb e, in più, MS-DOS WINDOWS 386 e mouse	**L. 9.990.000**

Amiga 2000	*Microprocessore Motorola 68000; clock 7,16 MHz; 1MB RAM; 256KB ROM; kickstart in* ROM. Uscita PAL; 4 uscite a 4 voci su 2 canali (stereofonia). Amiga DOS; Amiga multitasking. 7 slots di sistema (2 combinati Amiga - Ibm PC AT); 1 slot video; 1 accesso diretto CPU 86 pin. Compatibilità MS-DOS interna con schede Janus (XT/AT CARD). Possibilità di un secondo drive interno da 3" 1/2 e di un drive interno da 5" 1/4. *Monitor a colori 1084* tastiera; mouse. Workbench; Extras. MONITOR ESCLUSO	***L. 2.005.000***
	Monitor	
2080	*Monitor a colori ad alta risoluzione e lunga persistenza 14" con audio; 640x400 pixel; 4096* colori per A500/A1000/A2000/PC AT	***L. 630.000***
1084	Monitor a colori ad alta risoluzione 14" con audio; antiriflesso; 640x400 pixel; 4096 colori per A500/A1000/A2000/PC AT	**L. 545.000**
1402	Monitor monocromatico da 12"	**L. 255.000**
1450	Monitor monocromatico BISYNC	**L. • □**
1950	Monitor colore BISYNC	**L. • □**
A 2024	Monitor monocromatico alta risoluzione a pagina bianca	**L. • □**
	Stampanti	
MPS 1200P	Stampante a 9 aghi 120 cps bidirezionale 80 colonne. Interfaccia Commodore per PC 10/III PC 40 PC/60/80	**L. 480.000**
MPS 1250	Stesse caratteristiche del modello MPS 1200P, interfacce seriale RS232C e parallela Centronics di serie	**L. 495.000**
MPS 1500	Stampante a colori a 9 aghi 130 cps; bidirezionale; 80 colonne; carta in modulo singolo e trascinamento modulo continuo; 4 colori; per Amiga e PC	
	Unità dischi	
K 910	Floppy disk drive aggiuntivo da 3" 1/2 e 1.44 Mb per PC 10-III e PC 20-III	**L. 295.000**
A 1010	Floppy disk drive esterno da 3" 1/2	**L. 395.000**
A 2010	Floppy disk drive interno da 3" 1/2	**L. 290.000**
A 2092 + PC 5060	Hard disk da 20 Mb settorizzabile PC/Amiga + scheda controller per hard disk per A 2000	**L. 845.000**
A 2080 + A 2020	Scheda Janus XT compatibile + drive	**L. 895.000**
A 2286 + A 2020	Scheda Janus II AT compatibile + drive	**L. 1.285.000**
A 2620	*Scheda 68020 con processore a 32 bit; 14,28 MHZ*	***L. •***
A 2090 + A 2092	Hard disk 20 Mb; interfaccia SCSI per A 2000	**L. 1.025.000**
A 2090 + A 2094	Hard disk 40 Mb; interfaccia SCSI per A 2000	**L. 1.575.000**
A 590	Hard disk da 20 Mb + fast RAM per A 500	**L. • □**
	Accessori	
PC EXP1	Box esterno per espansione PCI; può contenere 3 schede aggiuntive e hard disk alimentazione e cavi compresi	**L. 590.000**
A 2052	Scheda di espansione RAM da 2 Mb per A 2000 (max 4)	**L. 590.000**
A 2060	Scheda video/RF molulatore per TV e VCR per A2000	**L. 138.000**
A 2058	Espansione di memoria da 8 Mb RAM per A2000	**L. 3.760.000**
A UNIX	Scheda UNIX utilizzabile con A2620 (68020 board)	**L. • □**
1352	*Mouse due tasti collegabile alla serie dei PC include Microsoft Mouse Driver*	***L. 72.000***
A 2300	Genlock card semiprofessionale per A2000; permette di miscelare grafici con videocamere VCR; selettore per video Amiga, monitors esterni, proiettori; include software per titolazione	**L. 345.000 Δ**
A 2350	Professional Video Adapter Card per A 2000. Genlock professionale con freeze frame e digitalizzatore video incluse software di controllo per la gestione interattiva	**L. 1.645.000 Δ**

• prezzo da definire
Δ disponibile da giugno '88
□ disponibile da settembre '88

I COMMODORE POINT

I primi 100 negozi selezionati dalla "Commodore" per la vendita dei suoi prodotti

Al Risparmio	*V.le Monza 204*	20128 Milano	Tel. 02-2573440
Braha Alberto S.D.F.	*Via Pier Capponi 5*	20145 Milano	Tel. 02-437468
E.D.S.	*C.so Porta Ticinese 4*	20123 Milano	Tel. 02-8322045
ESC	*Via Roggia Scagna 7*	20127 Milano	Tel. 02-2871300
Faref	*Via A. Volta 21*	20121 Milano	Tel. 02-650042
Giglioni	*Via D'Ovidio 8*	20131 Milano	Tel. 02-2360397
Giglioni	*V.le Luigi Sturzo 45*	20154 Milano	Tel. 02-654906
Logitek	*Via Golgi 60*	20133 Milano	Tel. 02-538931
Marcucci	*Via F.lli Bronzetti 37*	20129 Milano	Tel. 02-7386051
Melchioni	*Via P. Colletta 37*	20135 Milano	Tel. 02-57941
Messaggerie Musicali	*Galleria Del Corso*	20100 Milano	Tel. 02-50841
Newel	*Via Mac Mahon 75*	20155 Milano	Tel. 02-323492
Rivola	*Via Vitruvio 43*	20124 Milano	Tel. 02-6694160
F.lli Galimberti	*Via Naz. dei Giovi 28/36*	20030 Barlassina (Mi)	Tel. 0362-560625
P. Giorgio Ostellari	*Via Molino Arese 65*	20031 Cesano Mad.(Mi)	Tel. 0362-504392
P. Giorgio Ostellari	*Via Milano 300*	20233 Desio (Mi)	Tel. 0362-621042
GBC Italiana	*V.le Matteotti 66*	20092 Cinisello B. (Mi)	Tel. 02-6181801/ 6189391
	Via Petrella 6	20100 Milano	Tel. 02-203607/8
	Via Cantoni 7	20100 Milano	Tel. 02-437478
Casa Della Musica	*Via Indipendenza 21*	20093 Cologno Mon.(Mi)	Tel. 02-2542117
Penati	*Via Verdi 28/30*	20011 Corbetta (Mi)	Tel. 02-9779401
EPM System	*V.le Italia 12*	20094 Corsico (Mi)	Tel. 02-4407979
Centro Comp. Pandolfi	*Via Corridoni 18*	20025 Legnano (Mi)	Tel. 0331-546426
Computeam	*Via Vecellio 41*	20035 Lissone (Mi)	Tel. 039-481010
Futura	*Via Solferino 31*	20075 Lodi (Mi)	Tel. 0371-54457
MBM Inf. Sys.	*Corso Roma 112*	20075 Lodi (Mi)	Tel. 0371-53610
L'Amico del computer	*V.le Lombardia 17*	20077 Melegnano (Mi)	Tel. 02-9838341
Bit 84	*Via Italia 4*	20052 Monza (Mi)	Tel. 039-320813
I.C.O.	*Via dei Tigli 14*	20090 Opera (Mi)	Tel. 02-5242146
Comif	*Via Autolinee 10*	24100 Bergamo	Tel. 035-218553
Cordani	*Via dei Caniana 8*	24100 Bergamo	Tel. 035-258184
D.R.B.	*Via Borgo Palazzo 65*	24100 Bergamo	Tel. 035-237292
New Systems	*Via Paglia 36*	24100 Bergamo	Tel. 035-248109
Computer Team hi-tec	*Via Verdi 1/B*	24030 Carvico (Bg)	Tel. 035-790244
Ott. Optometr. Rovetta	*P.zza Garibaldi 6*	24065 Lovere (Bg)	Tel. 035-960705
A.I.S. International	*Via San Carlo 25*	24016 S.Pellegrino T.(Bg)	Tel. 0345-22662
Sisthema	*Via Roma 45*	24067 Sarnico (Bg)	Tel. 035-910750
Computer Center	*Via Cipro 62*	25125 Brescia	Tel. 030-223230
Informatica 2000	*Via Stazione 16/B*	25100 Brescia	Tel. 030-54015
Vigasio Mario	*Portici Zanardelli 3*	25100 Brescia	Tel. 030-59330/ 295858
Mister Bit	*Via Mazzini 70*	25043 Breno (Bs)	Tel. 0364-22835
Cavalli Pietro	*Via 10 Giornate 14B*	25030 Castrezzato (Bs)	Tel. 030-714013
Vietti Giuseppe	*Via Milano 1/B*	25032 Chiari (BS)	Tel. 030-7100206
Megabyte	*P.zza Duomo 17*	25015 Desenzano G.(Bs)	Tel. 030-9144880
Ditta Resi Rino	*Via XX Settembre 7*	25016 Ghedi (BS)	Tel. 030-901224
Info Cam	*Provinciale 38*	25050 Gratacasolo (Bs)	Tel. 0364-89379
Il Computer	*Via Indipendenza 90*	22100 Como	Tel. 031-240959
2M Elettronica	*Via Sacco 3*	22100 Como	Tel. 031-278227
Eletrongros	*Via L. Da Vinci 54*	22062 Barzanò (Co)	Tel. 039-957318
Ega	*Via Mazzini 42*	22065 Cassago Br.(Co)	Tel. 039-956307
Ega	*Via Aldo Moro 17*	22043 Galbiate (Co)	Tel. 0341-522028
Data Found Comp. Sh.	*Via A. Volta 4*	22036 Erba (Co)	Tel. 031-645761
Righi Elettronica	*Via G. Leopardi 26*	22077 Olgiate Com. (Co)	Tel. 031-946766
Fumagalli	*Via Cairoli 48*	22053 Lecco (Co)	Tel. 0341-863341
Cima Eelettronica	*Via Leonardo Da Vinci 7*	22053 Lecco (Co)	Tel. 0341-371106
Mondo Computer	*Via Giuseppina 11/B*	26100 Cremona	Tel. 0372-882079
Prisma	*Via Buoso Da Dovara 8*	26100 Cremona	Tel. 0372-437900
Telco	*P.zza Marconi 2/A*	26100 Cremona	Tel. 0372-31544
Elcom/GBC	*Via IV Novembre 56/58*	26013 Crema (Cr)	Tel. 0373-83393
Euroelettronica	*Via XX Settembre*	26013 Crema (Cr)	Tel. 0373-86966
Kofbaker	*Via Marchi 65/B*	26039 Vescovato (CR)	Tel. 0372-818601
Computer	*Galleria Fermi 7*	46100 Mantova	Tel. 0376-325616
32 Bit (Comp. St.)	*Via Cesare Battisti 14*	46100 Mantova	Tel. 0376-326770
Elettronica di Basso	*V.le Risorgimento 69*	46100 Mantova	Tel. 0376-329311
Poliware	*C.so C. Alberto 76*	27100 Pavia	Tel. 0382-22636
Log. Infor. Com. Shop	*V.le Monte Grappa 32*	27029 Vigevano (Pv)	Tel. 0381-81883
M. Visentin	*C.so Vitt. Emanuele 76*	27029 Vigevano (Pv)	Tel. 0381-83833
Computer Line	*Via G. Carducci 4*	29100 Piacenza	Tel. 0523-30691
Delta Computer	*Via Mart. della Resistenza 15/4*	29100 Piacenza	Tel. 0523-753318
Sover	*Via IV Novembre 60*	29100 Piacenza	Tel. 0523-34388
Cipolla Mauro	*Via Tremogge 25*	23100 Sondrio	Tel. 0342-213569
Fotonova		23010 S. Pietro Ber. (So)	Tel. 0342-492319
Dimeco Sistemi	*Via Garibaldi*	21100 Varese	Tel. 0332-237201
Il Centro Elettronico	*Via Morazzone 2*	21100 Varese	Tel. 0332-231006
Supergames	*Via Carrobbio 13*	21100 Varese	Tel. 0332-241092
Busto Bit	*Via Gavinana 17*	21052 Busto Arsizio (Va)	Tel. 0331-625034
Crespi Giuseppe & C.	*V.le Lombardia 59*	21053 Castellanza (Va)	Tel. 0331-503023
Computer Shop	*Via A. Da Brescia 2*	21013 Gallarate (Va)	Tel. 0331-798612
Lima Import-Export	*Via Clerici 196*	21040 Gerenzano (Va)	Tel.
J.A.C. nuove tecn.	*Via Matteotti 38*	21018 Sesto Cal. (Va)	Tel. 0331-923134
Bit Micro	*Via Mazzini 102*	15100 Alessandria	Tel. 0131-443252
West Records	*C.so Roma 85*	15100 Alessandria	Tel. 0131-441090
S.G.E. Elettronica	*Via Bandello 19*	15057 Tortona (Al)	Tel. 0131-867709
Record di Forina G.	*C.so Alfieri 166/3*	14100 Asti	Tel. 0141-34240
Il Diagramma Italia	*P.le Libertà 4*	12100 Cuneo	Tel. 0171-57067
Rossi Computers	*C.so Nizza 42*	12100 Cuneo	Tel. 0171-2339
Punto Bit	*C.so Langhe 26/C*	12051 Alba (Cn)	Tel. 0173-49833
SDI	*Via Vitt. Emanuele 250*	12042 BRA (CN)	Tel. 0172-421392
Aschieri Gianfranco	*C.so Eman. Filiberto 6*	12045 Fossano (CN)	Tel. 0172-62995
Curetti Augusto	*C.so Italia 3*	12054 Mondovì (Cn)	Tel. 0174-42014
Ditta Elettrogamma	*C.so Risorgimento 20*	28100 Novara	Tel. 0321-176358
Elcom	*C.so Mazzini 11*	28100 Novara	Tel. 0321-391293
Programma 3	*V.le Buonarroti 8*	28100 Novara	Tel. 0321-36367/ 399903
Punto Video	*C.so Risorgimento 391*	28100 Novara	Tel. 0321-477367
Computer	*Via Monte Zeda 4*	28041 Arona (No)	Tel.
All Computer	*C.so Garibaldi 106*	28021 Borgomanero (No)	Tel. 0322-844142
Micrologic	*Via Giovanni XIII 2*	28037 Domodossola (No)	Tel.
Elliott Computer Shop	*Via Don Minzoni 32*	28044 Intra (No)	Tel. 0323-43517
ABA Elettronica	*Via C. Fossati 5/P*	10100 Torino	Tel. 011-302065
Alex Computer e giochi	*C.so Francia 333/4*	10142 Torino	Tel. 011-7730184
C.D.M. Elettronica	*Via Marochetti 17*	10126 Torino	Tel. 011-636345/ 634900
Computing New	*Via Marco Polo 40/E*	10129 Torino	Tel. 011-501512
De Bug	*C.so Vitt. Emanuele II 22*	10100 Torino	Tel. 011-832986
Desme Universal	*Via San Secondo 95*	10100 Torino	Tel. 011-592551
F.D.S.	*Via Borgaro 86/D*	10100 Torino	Tel. 011-2168900
Computer Home	*Via San Donato 46/D*	10100 Torino	Tel. 011-4731196
Informatica Italia	*C.so Re Umberto 129*	10128 Torino	Tel. 011-501647
MT Informatica	*C.so Giulio Cesare 58*	10100 Torino	Tel. 011-850955/ 238803
New Bus. Computer	*Via Nizza 45*	10100 Torino	Tel. 011-214235
Radio Tv Mirafiori	*C.so Un. Sovietica 381*	10135 Torino	Tel. 011-616190/ 6197189
SMT Elettronica	*Via Bibiana 83/B*	10147 Torino	Tel. 011-218243
Paul e Chico Videos.	*Via Vitt. Emanuele 52*	10023 Chieri (To)	Tel. 011-9470295
Bit Informatica	*Via V. Emanuele 154*	10073 Ciriè (To)	Tel. 011-9205455
HI-FI Club	*C.so Francia 92C*	10093 Collegno (To)	Tel. 011-4110256
ICS	*Stradale Torino 73*	10015 Ivrea (To)	Tel. 0125-251797
Cerrutti Mauro	*C.so Torino 234*	10064 Pinerolo (To)	Tel. 0121-70316
Eurex	*C.so Indipendenza 5*	10086 Rivarolo C.se (To)	Tel. 0124-27984
Ditta Elettrogamma	*C.so Bormida*	13100 Vercelli	Tel. 0161-53689
Elettronica	*Strada Torino 15*	13100 Vercelli	Tel. 0161-393163
C.S.I. Teorema	*Via Losana 9*	13051 Biella (Vc)	Tel. 015-28622
Fotostudio Trevisan	*Via XXV Aprile 24/B*	13014 Cossato (VC)	Tel. 015-921431
Studio Fotogr. Imarisio	*P.zza Martiri Libertà 7*	13039 Trino (Vc)	Tel. 0161-82081
F.lli Gatti	*Via Festaz 75*	11100 Aosta	Tel. 0165-35659

PRODOTTI SYSTEMS EDITORIALE

Software su cassetta

La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.16000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.10000

Software su disco

Ms-Dos & Gw-Basic	L.25000
Ms-Dos & Gw-Basic (con prova di acquisto cassetta)	L.15000
La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.24000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.20000
Graphic Expander C/128 in modo 80 colonne	L.27000
Linguaggio macchina + Routine grafiche	L.12000

Offerta "Commodore speciale L.M." + dischetto — L.16000

Libri

64 programmi per il Commodore 64	L. 4800
I miei amici C/16 e Plus/4	L. 7000
Strategie vincenti per Commodore 64	L. 5800
62 programmi per Vic 20, C/16 e Plus/4	L. 6500
Utilities e giochi didattici	L. 6500
Tutti i segreti dello Spectrum	L. 7000
Simulazioni e test per la didattica	L. 7000
Impara giocando il Basic dello Spectrum	L. 7000
Micro Pascal per Commodore 64/128	L. 7000
Dal registratore al drive del C/64	L. 7000
Ada	L. 5000
Il linguaggio Pascal	L. 5000

Directory

Ciascun dischetto	L.12000

Arretrati

Ciascun numero arretrato di Commodore Computer Club	L. 5000
Ciascun numero arretrato di Personal Computer	L. 5000
Ciascun numero arretrato di VR Videoregistrare	L. 5000

Per un ottimale utilizzo del software "Matematica finanziaria" è opportuna la lettura degli articoli relativi pubblicati sui N.13, 14, 15 della rivista "Commodore" e sui N.1, 2 e 3 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Analisi di Bilancio" è opportuna la lettura degli articoli relativi pubblicati sui N.2, 3, 5 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Linguaggio Macchina e Routine grafiche per C/64" è opportuna la lettura del fascicolo "Commodore Speciale" appositamente dedicato.

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, la cifra di L.3000 per spese di imballo e spedizione, oppure L.6000 se si preferisce la spedizione per mezzo raccomandata.

Sconti e agevolazioni

Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L.50000 (di listino).

Gli abbonati hanno diritto allo sconto del 10% e alla spedizione gratuita se la somma totale raggiunge la cifra di L.50000 (di listino).

Oltre alla spedizione gratuita, viene praticato uno sconto del 10% (per gli abbonati è del 20%) se la cifra raggiunta per ciascun ordine raggiunge le L.100000 (di listino).

Abbonamenti

Commodore Computer Club (11 fascicoli)	L.45000
Personal Computer (11 fascicoli)	L.40000
Commodore Computer Club + Personal Computer (11 + 11 fascicoli)	L.70000
VR Videoregistrare (12 numeri)	L.45000

N.B.: la cifra per gli abbonamenti non può essere conteggiata per ottenere gli sconti e le agevolazioni di cui sopra.

Non è assolutamente possibile inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche il materiale desiderato.

In ogni caso sarebbe opportuno inviare la presente scheda, debitamente compilata, allegando la fotocopia della ricevuta del versamento effettuato.

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a: Systems Editoriale - Milano.

IN EDICOLA

N. 5 - L. 12.000

Commodore 64 Club

- Cover
- Terrore a Dunwich
- Hacker
- Lister
- Amadeus' Revenge
- Midnight
- Hawk mission
- Dispenser

4 GAMES
2 UTILITY
3 ADVENTURE

Dischetto a 2 facce, oltre 300 Kbyte di software

Commodore Club - Dir. Resp. A. Ronchetti Edizioni Systems Editoriale Srl - V.le Famagosta 75 -20142 Milano - Reg. Trib. MI. n. 104 del 25/2/84 - Distr. MePe.